

1 Przedmioty obowiązkowe

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Analiza matematyczna Calculus
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DOAM
6.	Rodzaj przedmiotu/modułu Obowiązkowy
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) pierwszy
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład – 60 godzin, ćwiczenia – 45 godzin, repetytorium – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Matematyka na poziomie szkoły średniej
13.	Cele przedmiotu Podstawowym celem przedmiotu jest zapoznanie studentów z podstawowymi narzędziami analizy, pojęciem pochodnej i całki, oraz z podstawowymi metodami argumentowania i wnioskowania.
14.	Treści programowe <ol style="list-style-type: none">1. Liczby rzeczywiste i zespolone.2. Ciągi i szeregi liczbowe rzeczywiste i zespolone, twierdzenie Bolzano-Weierstrassa, kryteria zbieżności szeregów, szeregi potęgowe.3. Funkcje jednej zmiennej, funkcje ciągłe, pochodna, wzór Taylora, podstawowe zastosowania, ekstrema.4. Całkowanie, funkcja pierwotna, całka oznaczona, zastosowania całek, podstawowe algorytmy numeryczne.5. Ciągi i szeregi funkcyjne, zbieżność jednostajna, zamiana kolejności operacji analitycznych, funkcje analityczne.

15.	Zakładane efekty uczenia się	
	Wiedza	
	Zna i rozumie schemat dowodu indukcyjnego	K_W01
	Zna i rozumie podstawowe pojęcia i twierdzenia rachunku różniczkowego jednej zmiennej	K_W01
	Rozumie rolę rachunku różniczkowego jednej zmiennej w rozwiązywaniu zagadnień optymalizacyjnych	K_W01
Umiejętności		
	Prowadzi proste dowody indukcyjne	K_U02
	Potrafi badać zbieżność ciągów oraz obliczać granice ciągów i funkcji stosując różne poznane metody	K_U01, K_U02
	Potrafi obliczać pochodne pierwszego i wyższych rzędów funkcji jednej zmiennej	K_U01
	Wykorzystuje metody rachunku różniczkowego funkcji jednej zmiennej w zagadnieniach optymalizacyjnych	K_U01
Kompetencje społeczne		
	Projektuje ścieżkę swojego kształcenia uwzględniając swoje mocne i słabe strony	K_K01
	Jest świadom ograniczeń własnej wiedzy i umiejętności	K_K01
16.	Literatura obowiązkowa i zalecana	
	<ul style="list-style-type: none"> • Kuratowski, K., Rachunek różniczkowy i całkowy • Fichtenholz, G., Rachunek różniczkowy i całkowy • Rudin, W., Podstawy analizy matematycznej 	
17.	Metody weryfikacji zakładanych efektów kształcenia	
	Pisemne kolokwia, pisemny egzamin końcowy	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Do zaliczenia ćwiczeń konieczne jest uzyskanie minimum punktów z 3 kolokwiiw śródsesestralnych. Zaliczenie ćwiczeń jest warunkiem dopuszczenia do egzaminu końcowego. Do zdania egzaminu końcowego wymagane jest uzyskanie minimum punktów.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	60 godz.
	ćwiczenia	45 godz.
	repetitorium	30 godz.
	Praca własna studenta	
	Przygotowanie się do egzaminu końcowego	30 godz.
	Przygotowanie się do ćwiczeń i kolokwiiw	80 godz.
	Sumarycznie	
	Łączna liczba godzin	245 godz.
	Liczba punktów ECTS	10

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Logika dla informatyków Logic for Computer Science
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DOLI
6.	Rodzaj przedmiotu/modułu Obowiązkowy
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) pierwszy
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin, repetytorium — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu –
13.	Cele kształcenia dla przedmiotu Wykształcenie umiejętności ścisłego formułowania myśli i poprawnego rozumowania. Zaznajomienie studentów z podstawowymi pojęciami i metodami logiki i teorii mnogości, przydatnymi w pracy informatyka.
14.	Treści programowe <ol style="list-style-type: none"> 1. Zasada indukcji. 2. Składnia i semantyka rachunku zdań i rachunku predykatów. Pojęcie spełniania i prawdziwości formuł. niesprzeczność zbioru formuł. 3. Podstawowe pojęcia teoriomnogościowe i operacje na zbiorach: suma, iloczyn, iloczyn kartezjański, zbiór potęgowy, relacje, funkcje, relacje równoważności, klasy abstrakcji, zbiór ilorazowy. 4. Moce zbiorów. Zbiory skończone i nieskończone. Zbiory przeliczalne i zbiory mocy continuum. Twierdzenia Cantora i Cantora-Bernsteina. 5. Porządki częściowe i liniowe. Dobrze porządki. Indukcja noetherowska. 6. Unifikacja termów. Informacja o metodzie rezolucji. 7. Dowodzenie twierdzeń. Informacja o systemie naturalnej dedukcji.

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 192 1161 320"> <tr> <td>zna wybrane, podstawowe pojęcia teoriomnogościowe (zbiory i operacje na zbiorach), pojęcie mocy zbioru, pojęcia porządków częściowych i liniowych</td> <td>K_W02</td> </tr> <tr> <td>zna rachunek zdań i predykatów, pojęcia spełnialności i prawdziwości oraz metody dowodzenia (indukcja, rezolucja, dedukcja)</td> <td>K_W02</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 405 1161 629"> <tr> <td>posługuje się rachunkiem zdań i kwantyfikatorów</td> <td>K_U02</td> </tr> <tr> <td>potrafi przeprowadzać ścisłe rozumowanie z wykorzystaniem matematycznych metod dowodzenia (indukcji, rezolucji, dedukcji)</td> <td>K_U02</td> </tr> <tr> <td>potrafi stosować pojęcia teoriomnogościowe do analizy i modelowania problemów w informatyce</td> <td>K_U02</td> </tr> <tr> <td>potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia danego tematu lub odnalezieniu brakujących elementów rozumowania</td> <td>K_U02</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 712 1161 779"> <tr> <td>rozumie i docenia znaczenie uczciwości intelektualnej w działaniach własnych i innych osób, postępuje etycznie</td> <td>K_K01, K_K02, K_K03</td> </tr> </table>	zna wybrane, podstawowe pojęcia teoriomnogościowe (zbiory i operacje na zbiorach), pojęcie mocy zbioru, pojęcia porządków częściowych i liniowych	K_W02	zna rachunek zdań i predykatów, pojęcia spełnialności i prawdziwości oraz metody dowodzenia (indukcja, rezolucja, dedukcja)	K_W02	posługuje się rachunkiem zdań i kwantyfikatorów	K_U02	potrafi przeprowadzać ścisłe rozumowanie z wykorzystaniem matematycznych metod dowodzenia (indukcji, rezolucji, dedukcji)	K_U02	potrafi stosować pojęcia teoriomnogościowe do analizy i modelowania problemów w informatyce	K_U02	potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia danego tematu lub odnalezieniu brakujących elementów rozumowania	K_U02	rozumie i docenia znaczenie uczciwości intelektualnej w działaniach własnych i innych osób, postępuje etycznie	K_K01, K_K02, K_K03				
zna wybrane, podstawowe pojęcia teoriomnogościowe (zbiory i operacje na zbiorach), pojęcie mocy zbioru, pojęcia porządków częściowych i liniowych	K_W02																		
zna rachunek zdań i predykatów, pojęcia spełnialności i prawdziwości oraz metody dowodzenia (indukcja, rezolucja, dedukcja)	K_W02																		
posługuje się rachunkiem zdań i kwantyfikatorów	K_U02																		
potrafi przeprowadzać ścisłe rozumowanie z wykorzystaniem matematycznych metod dowodzenia (indukcji, rezolucji, dedukcji)	K_U02																		
potrafi stosować pojęcia teoriomnogościowe do analizy i modelowania problemów w informatyce	K_U02																		
potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia danego tematu lub odnalezieniu brakujących elementów rozumowania	K_U02																		
rozumie i docenia znaczenie uczciwości intelektualnej w działaniach własnych i innych osób, postępuje etycznie	K_K01, K_K02, K_K03																		
16.	<p>Literatura obowiązkowa i zalecana</p> <ol style="list-style-type: none"> 1. Wojciech Guzicki, Piotr Zakrzewski, Wykłady ze wstępu do matematyki. Wprowadzenie do teorii mnogości, PWN, Warszawa 2005. 2. Wojciech Guzicki, Piotr Zakrzewski, Wstęp do matematyki. Zbiór zadań, PWN, Warszawa 2005. 3. Kazimierz Kuratowski, Wstęp do teorii mnogości i topologii, PWN, Warszawa 2004. 4. Wiktor Marek, Janusz Onyszkiewicz, Elementy logiki i teorii mnogości w zadaniach, PWN, Warszawa 2005. 5. Helena Rasiowa, Wstęp do matematyki współczesnej, PWN, Warszawa 2007. 6. Jerzy Tiuryn, Wstęp do teorii mnogości i logiki, Skrypt Uniw. Warszawskiego, 1994. 																		
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>prezentacja rozwiązania zadania, egzamin, sprawdziany/kolokwia</p>																		
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Ocena z ćwiczeń jest wystawiana na podstawie uzyskanej liczby punktów. Punkty otrzymuje się za przygotowane rozwiązania zadań (na podstawie deklaracji weryfikowanych w trakcie ćwiczeń), kartkówki i kolokwia.</p> <p>Aby zaliczyć wykład należy zdobyć minimalną wymaganą liczbę punktów z końcowego egzaminu piśmennego. Szczegółowe warunki zaliczenia przedmiotu są określone w regulaminie przedmiotu.</p>																		
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 1496 1428 1592"> <tr> <td>wykład</td> <td>30 godz.</td> </tr> <tr> <td>repetitorium</td> <td>30 godz.</td> </tr> <tr> <td>ćwiczenia</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 1675 1428 1803"> <tr> <td>przygotowanie do ćwiczeń</td> <td>45 godz.</td> </tr> <tr> <td>przygotowanie do egzaminu</td> <td>15 godz.</td> </tr> <tr> <td>studiowanie tematyki wykładów i literatury</td> <td>15 godz.</td> </tr> <tr> <td>przygotowanie do sprawdzianów/kolokwiów</td> <td>25 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 1883 1428 1951"> <tr> <td>Łączna liczba godzin</td> <td>190 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>8</td> </tr> </table>	wykład	30 godz.	repetitorium	30 godz.	ćwiczenia	30 godz.	przygotowanie do ćwiczeń	45 godz.	przygotowanie do egzaminu	15 godz.	studiowanie tematyki wykładów i literatury	15 godz.	przygotowanie do sprawdzianów/kolokwiów	25 godz.	Łączna liczba godzin	190 godz.	Liczba punktów ECTS	8
wykład	30 godz.																		
repetitorium	30 godz.																		
ćwiczenia	30 godz.																		
przygotowanie do ćwiczeń	45 godz.																		
przygotowanie do egzaminu	15 godz.																		
studiowanie tematyki wykładów i literatury	15 godz.																		
przygotowanie do sprawdzianów/kolokwiów	25 godz.																		
Łączna liczba godzin	190 godz.																		
Liczba punktów ECTS	8																		

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Algebra Algebra
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DOALG
6.	Rodzaj przedmiotu/modułu Obowiązkowy
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) pierwszy
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 45 godzin, ćwiczenia – 30 godzin, repetytorium – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu <ul style="list-style-type: none"> • Podstawy logiki i teorii mnogości (pojęcie relacji i rachunku zbiorów, formuły z kwantyfikatorami). • Podstawy analizy matematycznej (pojęcie wielomianu, całki, pochodnej).
13.	Cele przedmiotu Celem przedmiotu jest zaznajomienie studentów z podstawowymi pojęciami algebry liniowej i algebry abstrakcyjnej: <ul style="list-style-type: none"> • przestrzeniami liniowymi, • macierzami, • układami równań, • grupami, • pierścieniami, • ciałami. <p>Nacisk położony będzie na związki z informatyką oraz na przykłady zastosowań w informatyce.</p>
14.	Treści programowe <ol style="list-style-type: none"> 1. Przestrzenie liniowe. Zbiory liniowo niezależne. Bazy. 2. Macierze i przekształcenia liniowe. Rząd macierzy. Algorytm eliminacji Gaussa. 3. Wyznaczniki. Własności wyznaczników. Rozwinięcie Laplace'a. 4. Równania liniowe. Zbiór rozwiązań układu równań liniowych. Wzory Cramera. 5. Wartości i wektory własne. Podprzestrzenie niezmiennicze. Wielomian charakterystyczny. 6. Iloczyn skalarny. Rzut ortogonalny. Izometrie i przekształcenia ortogonalne. 7. Elementy geometrii. 8. Grupy — podstawowe pojęcia: rząd grupy, rząd elementu grupy, podgrupa. 9. Grupy permutacji. Rozkład permutacji na cykle. Znak permutacji. 10. Działanie grupy na zbiorze. Orbita i stabilizatory. Lemat Burnside'a. Warstwy. Twierdzenie Lagrange'a. 11. Homomorfizmy grup. Kongruencje. Dzielniki normalne. Grupa ilorazowa. 12. Arytmetyka modularna. Relacja podzielności. Pierścienie i pierścienie Zn. 13. Algorytm Euklidesa. Chińskie twierdzenie o resztach. 14. Pierścienie wielomianów. Podzielność wielomianów. 15. Przykład konstrukcji ciała skończonego.

15.	Zakładane efekty uczenia się	
	Wiedza	
	Zna podstawowe koncepcje algebry liniowej, w tym pojęcia przestrzeni liniowej, wektora, przekształcenia liniowego i macierzy oraz metody rozwiązywania układów równań liniowych.	K_W01
	Zna pojęcia grupy, pierścienia (w szczególności pierścienia wielomianów), ciała. Zna podstawowe ciała (Z_n), pierścienie i grupy.	K_W01 K_W01
	Umiejętności	
	Potrafi używać rachunku wektorowego i macierzowego: dodawać, mnożyć wektory i macierze. Potrafi zapisać przekształcenie liniowe jako macierz.	K_U01
	Potrafi rozwiązać układ równań liniowych, obliczyć wyznacznik, wyliczyć wartości własne macierzy.	K_U01
	Potrafi zmienić bazę przestrzeni liniowej, dokonać ortonormalizacji bazy.	K_U01
	Potrafi prowadzić obliczenia w grupach i pierścieniach, ze szczególnym uwzględnieniem pierścienia wielomianów i ciał Z_p .	K_U01, K_U02
	Potrafi wyznaczyć grupy permutacji obiektu kombinatorycznego.	K_U01, K_U02
	Kompetencje społeczne	
	Potrafi prosto, zrozumiale i przekonująco przedstawić rozumowanie matematyczne, zarówno pisemnie jak i ustnie.	K_K01
16.	Literatura obowiązkowa i zalecana <ol style="list-style-type: none"> 1. A. Białynicki-Birula, Algebra 2. G. Birkhoff, S. Mac Lane, Przegląd algebry współczesnej 3. B. Gleichgewicht, Algebra 4. A. Kostrikin, J. Manin Algebra liniowa i geometria 5. A. Kostrikin, Zbiór zadań z algebry 6. A. Kostrikin, Wstęp do algebra t. 1-3 7. A. Mostowski, M. Stark, Elementy algebry wyższej 8. A. Mostowski, M. Stark, Algebra liniowa 9. J. Rutkowski, Algebra abstrakcyjna w zadaniach 	
17.	Metody weryfikacji zakładanych efektów kształcenia <ul style="list-style-type: none"> • prezentacja zadań przy tablicy • pisemne prace domowe • egzamin 	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Wykład: egzaminy — egzamin połówkowy i egzamin na koniec semestru. Zadania sprawdzające znajomość podstawowych definicji, sposobów postępowania i umiejętność stosowania ich w konkretnym przypadku. Ćwiczenia: punkty za zadania z list zadań oraz prezentację przy tablicy, pisemne opracowanie zadania prezentowanego w czasie ćwiczeń.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	45 godz.
	ćwiczenia	30 godz.
	repetitorium	30 godz.
	Praca własna studenta	
	studiowanie tematyki wykładów i literatury	40 godz.
	przygotowanie do egzaminu	10 godz.
	przygotowanie do ćwiczeń	30 godz.
	egzamin połówkowy + egzamin	6 godz.
	Sumarycznie	
	Łączna liczba godzin	191 godz.
	Liczba punktów ECTS	8

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Metody programowania Programming methodology
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DOMP
6.	Rodzaj przedmiotu/modułu Obowiązkowy
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) pierwszy
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 45 godzin, ćwiczenio-pracownia — 45 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty <ul style="list-style-type: none"> • Logika dla informatyków • sugerowany Wstęp do informatyki Wymagane kompetencje <ul style="list-style-type: none"> • Podstawowa umiejętność wnioskowania logicznego • Wskazana bierna znajomość języka angielskiego
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest przygotowanie studentów do używania języków programowania jako formalnych narzędzi komunikacji między ludźmi, w szczególności jako technik, które mogą i powinny być stosowane do zapanowania nad złożonością intelektualną dużych systemów. W tym celu zajmiemy się zarówno konkretnymi narzędziami, których projektanci języków programowania dostarczają programistom, jak i ogólnymi zasadami konstrukcji języków programowania.

14.	<p>Treści programowe</p> <ol style="list-style-type: none"> 1. Elementy programowania 2. Rekursja i abstrakcja proceduralna 3. Wprowadzenie do abstrakcji danych 4. Dane hierarchiczne i własność domknięcia 5. Dane symboliczne i wielorakie reprezentacje danych 6. Składnia abstrakcyjna i podstawy ewaluacji 7. Środowiskowy model obliczeń 8. Procedury i domknięcia 9. Reprezentacja wartości w metajęzyku 10. Ewaluator metacykliczny 11. Zarządzanie złożonością: kontrakty 12. Zarządzanie złożonością: systemy typów 13. Obliczenia z danymi modyfikowalnymi 14. Elementy programowania współbieżnego 15. Programowanie w logice 														
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 719 1426 913"> <tr> <td>zna wybrane modele obliczania wartości programów i podstawowe metody wnioskowania o ich poprawności</td> <td>K_W05</td> </tr> <tr> <td>rozumie pojęcie i cel stosowania abstrakcji danych, a także sposoby budowania indukcyjnych struktur danych</td> <td>K_W05, K_W06</td> </tr> <tr> <td>ma przeglądową wiedzę na temat sposobów zarządzania złożonością większych programów</td> <td>K_W05, K_W06</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 994 1426 1160"> <tr> <td>potrafi przedstawić rozwiązania problemów używając formalizmu funkcyjnego języka programowania</td> <td>K_U03</td> </tr> <tr> <td>potrafi zaprojektować reprezentację danych odpowiednią dla rozważanego problemu</td> <td>K_U03</td> </tr> <tr> <td>potrafi zadać semantykę prostego języka poprzez zdefiniowanie ewaluatora</td> <td>K_U03</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 1240 1426 1308"> <tr> <td>rozumie znaczenie języków programowania jako narzędzia precyzyjnej komunikacji międzyludzkiej</td> <td>K_K01, K_K02, K_K04</td> </tr> </table>	zna wybrane modele obliczania wartości programów i podstawowe metody wnioskowania o ich poprawności	K_W05	rozumie pojęcie i cel stosowania abstrakcji danych, a także sposoby budowania indukcyjnych struktur danych	K_W05, K_W06	ma przeglądową wiedzę na temat sposobów zarządzania złożonością większych programów	K_W05, K_W06	potrafi przedstawić rozwiązania problemów używając formalizmu funkcyjnego języka programowania	K_U03	potrafi zaprojektować reprezentację danych odpowiednią dla rozważanego problemu	K_U03	potrafi zadać semantykę prostego języka poprzez zdefiniowanie ewaluatora	K_U03	rozumie znaczenie języków programowania jako narzędzia precyzyjnej komunikacji międzyludzkiej	K_K01, K_K02, K_K04
zna wybrane modele obliczania wartości programów i podstawowe metody wnioskowania o ich poprawności	K_W05														
rozumie pojęcie i cel stosowania abstrakcji danych, a także sposoby budowania indukcyjnych struktur danych	K_W05, K_W06														
ma przeglądową wiedzę na temat sposobów zarządzania złożonością większych programów	K_W05, K_W06														
potrafi przedstawić rozwiązania problemów używając formalizmu funkcyjnego języka programowania	K_U03														
potrafi zaprojektować reprezentację danych odpowiednią dla rozważanego problemu	K_U03														
potrafi zadać semantykę prostego języka poprzez zdefiniowanie ewaluatora	K_U03														
rozumie znaczenie języków programowania jako narzędzia precyzyjnej komunikacji międzyludzkiej	K_K01, K_K02, K_K04														
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • H. Abelson, G.J. Sussman, J. Sussman, Struktura i interpretacja programów komputerowych, WNT, Warszawa 2002, https://mitpress.mit.edu/sicp/ (obowiązkowa) • D.P. Friedman, M. Felleisen, The Little Schemer, MIT Press, 1995, https://mitpress.mit.edu/books/little-schemer (uzupełniająca) • M. Felleisen, R. Findler, M. Flatt, S. Krishnamurthi, How to Design Programs, MIT Press, 2018, http://www.htdp.org/ (uzupełniająca) • F. Bice, R. DeMaio, S. Florence, M. Lin, S. Lindeman, N. Nussbaum, E. Peterson, R. Plessner, D. Van Horn, M. Felleisen, C. Barski, Realm of Racket, no starch press, 2013, http://realmofracket.com/ (uzupełniająca) 														
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>egzamin pisemny lub pisemno-ustny, kolokwium, napisanie programu komputerowego, prezentacja rozwiązania zadania</p>														
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Do zaliczenia ćwiczenie-pracowni należy:</p> <ul style="list-style-type: none"> • uzyskać wymaganą regulaminem przedmiotu liczbę punktów z kolokwium zaliczeniowego; • uzyskać wymaganą regulaminem przedmiotu liczbę punktów za rozwiązania programistycznych zadań domowych; • uzyskać wymaganą regulaminem przedmiotu liczbę zaliczonych obecności na ćwiczeniach. <p>Do zaliczenia niezbędne jest jednoczesne spełnienie wszystkich powyższych warunków. Egzamin ma formę pisemną lub pisemno-ustną, ustaloną regulaminem przedmiotu.</p>														

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	45 godz.
	ćwiczenio-pracownia	45 godz.
	Praca własna studenta	
	Konsultacje: w ramach indywidualnych potrzeb	0 godz.
	Rozwiązanie programistycznych zadań domowych	60 godz.
	Przygotowanie do ćwiczeń	40 godz.
	Udział w kolokwiach i egzaminie	5 godz.
	Utrwalenie materiału i przygotowanie do egzaminu	20 godz.
Sumarycznie		
Łączna liczba godzin	215 godz.	
Liczba punktów ECTS	9	

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Analiza numeryczna Numerical analysis
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DOANM
6.	Rodzaj przedmiotu/modułu Obowiązkowy
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) drugi
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 45 godzin, ćwiczenia — 30 godzin, repetytorium — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu <ol style="list-style-type: none"> Analiza matematyczna (ciągły, funkcje, granice, szeregi, różniczkowanie, szereg Taylora, całkowanie). Algebra liniowa (algebra macierzy, układy równań liniowych, pojęcie bazy, jej wymiar i ortogonalizacja). Umiejętność programowania w języku umożliwiającym obliczenia zmiennopozycyjne.
13.	Cele kształcenia dla przedmiotu Celem zajęć jest przedstawienie podstawowych metod i algorytmów rozwiązywania typowych zadań obliczeniowych. Omawiane zagadnienia mają wielorakie zastosowania m.in. w obliczeniach naukowych czy grafice komputerowej.
14.	Treści programowe <ol style="list-style-type: none"> Analiza błędów. Arytmetyka numeryczna. Uwarunkowanie zadania. Algorytmy numerycznie poprawne. Rozwiązywanie równań nieliniowych. Ogólna teoria metod iteracyjnych. Metody: bisekcji, Newtona i siecznych. Interpolacja. Wzór interpolacyjny Lagrange'a. Reszta wzoru interpolacyjnego. Wzór interpolacyjny Newtona. Interpolacja Hermite'a. Interpolacja za pomocą funkcji sklepanych III stopnia. Krzywe Béziera i ich zastosowanie w grafice komputerowej. Aproksymacja. Aproksymacja średniokwadratowa za pomocą wielomianów - wielomiany ortogonalne, twierdzenie o n-tym wielomianie optymalnym. Aproksymacja jednostajna - twierdzenie o alternansie, informacja o algorytmie Remez'a konstrukcji wielomianu optymalnego, wielomiany prawieoptymalne. Kwadratury. Kwadratura liniowa. Reszta i rząd kwadratury. Zbieżność ciągu kwadratur. Kwadratury interpolacyjne. Kwadratury Newtona-Cotesa. Wzory złożone: trapezów i Simpsona. Metoda Romberga. Kwadratury Gaussa. Rozwiązywanie układów równań liniowych. Uwarunkowanie zadania. Rozkład macierzy kwadratowej na iloczyn macierzy trójkątnych. Metoda eliminacji Gaussa. Iteracyjne metody rozwiązywania układów równań liniowych.

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <p>zna podstawowe pojęcia, metody i algorytmy numeryczne (arytmetyka zmienopozycyjna, teoria błędów, interpolacja i aproksymacja funkcji, rozwiązywanie równań nieliniowych, całkowanie numeryczne, metody numeryczne algebry liniowej)</p> <p>zna numeryczne podstawy grafiki komputerowej (krzywe sklejące, krzywe Béziera)</p> <p>Umiejętności</p> <p>potrafi zaimplementować poznane algorytmy numeryczne i stosować je do rozwiązywania podstawowych zadań matematyki obliczeniowej</p> <p>potrafi samodzielnie analizować wybrane zagadnienia, dyskutować i prezentować wnioski przed grupą</p> <p>potrafi wykorzystać zdobytą wiedzę z zakresu analizy numerycznej do formułowania i rozwiązywania zadań obliczeniowych stosując metody analityczne, symulacyjne i eksperymentalne oraz interpretować uzyskane wyniki i wyciągać wnioski</p> <p>potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia zagadnień z zakresu analizy numerycznej lub odnalezieniu brakujących elementów rozumowania</p> <p>Kompetencje inżynierskie</p> <p>potrafi planować i przeprowadzać eksperymenty z zakresu szeroko rozumianej analizy numerycznej lub grafiki komputerowej, w tym pomiary i symulacje komputerowe, interpretować uzyskane wyniki i wyciągać wnioski</p> <p>Kompetencje społeczne</p> <p>jest świadom ograniczeń własnej wiedzy i umiejętności w zakresie analizy numerycznej, adekwatnie ocenia poziom swoich kompetencji, zna swoje mocne i słabe strony oraz rozumie konieczność doskonalenia w tym zakresie</p> <p>jest świadom roli i znaczenia wiedzy z zakresu analizy numerycznej w rozwiązywaniu problemów o charakterze poznawczym i praktycznym, w tym dla funkcjonowania współczesnego społeczeństwa, zna niebezpieczeństwa z tym związane</p>	<p>K_W04</p> <p>K_W04</p> <p>K_U05, Inż_U01</p> <p>K_U02</p> <p>K_U05, Inż_U01</p> <p>K_U02, K_U05</p> <p>Inż_U01</p> <p>K_K01</p> <p>K_K02, K_K04</p>
16.	<p>Literatura obowiązkowa i zalecana</p> <p>Literatura</p> <ul style="list-style-type: none"> • A. Björck, G. Dahlquist, Metody numeryczne, PWN, 1987. • M. Dryja, J.i M. Jankowscy, Przegląd metod i algorytmów numerycznych, cz. 1 i 2, WNT, 1988. • D. Kincaid, W. Cheney, Analiza numeryczna, WNT, 2005. • G. Dahlquist, A. Björck, Numerical Methods in Scientific Computing, Vol. I, SIAM, 2008. • W. Gautschi, Numerical Analysis. An Introduction, Birkhäuser, 1997. • G. Hämmerlin, K.-H. Hoffmann, Numerical Mathematics, Springer-Verlag, 1991. • A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics, Springer- Verlag, 2000. 	
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <ul style="list-style-type: none"> • egzamin pisemny, • sprawdzian pisemny, • w ramach ćwiczeń z systemem deklaracji: prezentacja i analiza rozwiązań zadanych problemów teoretycznych i programistycznych (zadania typu włącz komputer) oraz kartkówki <p>(wszystkie formy w zakresie treści programowych z uwzględnieniem zakładanych efektów uczenia się)</p>	
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Ocena z przedmiotu zależy od liczby punktów zdobytych z egzaminu pisemnego (waga 0,75) oraz z ćwiczeń (waga 0,25). Minimalne wymagania: zaliczenie ćwiczeń na ocenę pozytywną (tj. zaliczenie każdego z trzech bloków list zadań + przynajmniej 45% punktów) i zdobycie przynajmniej 40% punktów z egzaminu pisemnego. Szczegółowe zasady zaliczenia ćwiczeń przeprowadzanych z wykorzystaniem systemu deklaracji oraz całego przedmiotu dostępne są na stronie wykładowcy i w systemie SKOS oraz omawiane są na pierwszych ćwiczeniach i pierwszym wykładzie.</p>	

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	45 godz.
	ćwiczenia	30 godz.
	repetytorium	30 godz.
	udział w sprawdzianach i egzaminie	5 godz.
	Praca własna studenta	
	przygotowanie do egzaminu	30 godz.
	przygotowanie do sprawdzianów/kolokwium	15 godz.
	studiowanie tematyki wykładów i literatury	30 godz.
	przygotowanie do ćwiczeń	20 godz.
Sumarycznie		
Łączna liczba godzin	205 godz.	
Liczba punktów ECTS	8	

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Matematyka dyskretna (L) Discrete Mathematics
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DOMDM
6.	Rodzaj przedmiotu/modułu Obowiązkowy
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) drugi
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 45 godzin, ćwiczenia — 45 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Ogólna kultura matematyczna lub <ul style="list-style-type: none"> • Logika dla informatyków • Analiza • Algebra
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest przedstawienie elementów matematyki, które są szczególnie użyteczne dla informatyków i nie są częścią standardowego wykładu algebry, analizy czy rachunku prawdopodobieństwa. W szczególności są to przydatne informatykowi elementy teorii liczb, kombinatoryki i teorii grafów. Wiedza w tym zakresie jest przydatna w konstruowaniu i analizie algorytmów, w tym weryfikacji poprawności i szacowaniu złożoności obliczeniowej.

14.	<p>Treści programowe</p> <p>A. Elementy Algebry i Teorii Liczb</p> <ul style="list-style-type: none"> • Funkcje całkowitoliczbowe, arytmetyka modularna, operacje sufit i podłoga zaokrąglania liczb rzeczywistych, algorytm mergesort. • Asymptotyka funkcji liczbowych z uwzględnieniem zastosowań w szacowaniu złożoności czasowej algorytmów. • Podzielność liczb, algorytm Euklidesa. • Liczby Fibonacciego. • Liczby pierwsze i względnie pierwsze. Rozkład na czynniki. Funkcja Eulera. Chińskie twierdzenie o resztach. Twierdzenie Eulera. <p>B. Kombinatoryka</p> <ul style="list-style-type: none"> • Rozmieszczenia, permutacje, kombinacje, podziały (zbioru, liczby), Lemat Burnside'a. • Metody generowania prostych obiektów kombinatorycznych. • Przykłady prostych problemów definiowanych rekurencyjnie. • Rozwiązywanie równań rekurencyjnych, funkcje tworzące. • Liczby Catalana. • Zasada włączania i wyłączania. <p>C. Teoria grafów</p> <ul style="list-style-type: none"> • Definicja i przykłady grafów, grafy pełne, dwudzielne skierowane, stopień wierzchołka. • Drogi i cykle w grafach: grafy spójne i dwudzielne. • Drzewa - równoważność różnych definicji. • Komputerowa reprezentacja grafów. • Metody BFS i DFS przeszukiwania grafów. • Minimalne drzewa rozpinające - algorytmy Kruskala i Prima-Dijkstry. • Przechodnie domknięcie: algorytmy Dijkstry i Warshalla. Złożoność problemu. • Cykle i drogi Eulera. • Cykle i drogi Hamiltona tw. Ore i wielomianowa redukcja problemu drogi do cyklu i odwrotnie. • Grafy planarne. Tw. Kuratowskiego i wzór Eulera. • Przepływy w sieciach. • Kolorowanie grafów: zastosowanie - planowanie sesji egzaminacyjnej. Algorytm sekwencyjny i twierdzenie o 5-kolorowaniu grafów planarnych. 																						
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="223 1321 1428 1512"> <tr> <td>zna podstawowe sposoby zliczania i wyliczania obiektów kombinatorycznych</td> <td>K_W03</td> </tr> <tr> <td>zna metody rozwiązywania równań rekurencyjnych</td> <td>K_W03</td> </tr> <tr> <td>zna podstawowe pojęcia, twierdzenia i algorytmy teorii liczb</td> <td>K_W03</td> </tr> <tr> <td>zna notację asymptotyczną i sposoby szacowania i porównywania funkcji liczbowych</td> <td>K_W03</td> </tr> <tr> <td>zna podstawowe pojęcia, twierdzenia i algorytmy teorii grafów</td> <td>K_W03</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="223 1601 1428 1758"> <tr> <td>potrafi zliczać i wylizzać obiekty kombinatoryczne</td> <td>K_U04</td> </tr> <tr> <td>potrafi rozwiązywać równania rekurencyjne</td> <td>K_U04</td> </tr> <tr> <td>potrafi szacować i porównywać złożoności czasowe i pamięciowe algorytmów</td> <td>K_U02, K_U04</td> </tr> <tr> <td>umie zastosować pojęcia i twierdzenia teorii liczb</td> <td>K_U02, K_U04</td> </tr> <tr> <td>potrafi stosować pojęcia, twierdzenia i algorytmy podstawowej teorii grafów</td> <td>K_U04</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="223 1848 1428 1904"> <tr> <td>potrafi przejrzyście (w mowie i na piśmie) przedstawić rozumowania z dziedziny matematyki dyskretniej</td> <td>K_K01, K_K02, K_K04</td> </tr> </table>	zna podstawowe sposoby zliczania i wyliczania obiektów kombinatorycznych	K_W03	zna metody rozwiązywania równań rekurencyjnych	K_W03	zna podstawowe pojęcia, twierdzenia i algorytmy teorii liczb	K_W03	zna notację asymptotyczną i sposoby szacowania i porównywania funkcji liczbowych	K_W03	zna podstawowe pojęcia, twierdzenia i algorytmy teorii grafów	K_W03	potrafi zliczać i wylizzać obiekty kombinatoryczne	K_U04	potrafi rozwiązywać równania rekurencyjne	K_U04	potrafi szacować i porównywać złożoności czasowe i pamięciowe algorytmów	K_U02, K_U04	umie zastosować pojęcia i twierdzenia teorii liczb	K_U02, K_U04	potrafi stosować pojęcia, twierdzenia i algorytmy podstawowej teorii grafów	K_U04	potrafi przejrzyście (w mowie i na piśmie) przedstawić rozumowania z dziedziny matematyki dyskretniej	K_K01, K_K02, K_K04
zna podstawowe sposoby zliczania i wyliczania obiektów kombinatorycznych	K_W03																						
zna metody rozwiązywania równań rekurencyjnych	K_W03																						
zna podstawowe pojęcia, twierdzenia i algorytmy teorii liczb	K_W03																						
zna notację asymptotyczną i sposoby szacowania i porównywania funkcji liczbowych	K_W03																						
zna podstawowe pojęcia, twierdzenia i algorytmy teorii grafów	K_W03																						
potrafi zliczać i wylizzać obiekty kombinatoryczne	K_U04																						
potrafi rozwiązywać równania rekurencyjne	K_U04																						
potrafi szacować i porównywać złożoności czasowe i pamięciowe algorytmów	K_U02, K_U04																						
umie zastosować pojęcia i twierdzenia teorii liczb	K_U02, K_U04																						
potrafi stosować pojęcia, twierdzenia i algorytmy podstawowej teorii grafów	K_U04																						
potrafi przejrzyście (w mowie i na piśmie) przedstawić rozumowania z dziedziny matematyki dyskretniej	K_K01, K_K02, K_K04																						

16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • Charles R.B. Wright, Kenneth A. Ross: Matematyka dyskretna • Ronald L. Graham, Donald E. Knuth, Oren Patashnik: Matematyka Konkretna • Witold Lipski: Kombinatoryka dla programistów • Victor Bryant: Aspekty kombinatoryki • Robin J. Wilson: Wprowadzenie do teorii grafów 														
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>Egzamin pisemny, prezentacja zadań przy tablicy</p>														
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Do zaliczenia ćwiczeń należy zdobyć wymaganą, podaną w regulaminie przedmiotu liczbę punktów za zadania ćwiczeniowe.</p> <p>Egzamin ma formę pisemną. Aby go zaliczyć konieczne jest zdobycie odpowiedniej liczby punktów.</p>														
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1"> <tr> <td>wykład</td> <td>45 godz.</td> </tr> <tr> <td>ćwiczenia</td> <td>45 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1"> <tr> <td>przygotowanie do egzaminu</td> <td>40 godz.</td> </tr> <tr> <td>rozwiązywanie zadań z list</td> <td>90 godz.</td> </tr> <tr> <td>udział w egzaminie</td> <td>4 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1"> <tr> <td>Łączna liczba godzin</td> <td>224 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>9</td> </tr> </table>	wykład	45 godz.	ćwiczenia	45 godz.	przygotowanie do egzaminu	40 godz.	rozwiązywanie zadań z list	90 godz.	udział w egzaminie	4 godz.	Łączna liczba godzin	224 godz.	Liczba punktów ECTS	9
wykład	45 godz.														
ćwiczenia	45 godz.														
przygotowanie do egzaminu	40 godz.														
rozwiązywanie zadań z list	90 godz.														
udział w egzaminie	4 godz.														
Łączna liczba godzin	224 godz.														
Liczba punktów ECTS	9														

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Algorytmy i struktury danych (L) Algorithms and Data Structures
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DOASDL
6.	Rodzaj przedmiotu/modułu Obowiązkowy
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) drugi
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 60 godzin, ćwiczenio-pracownia — 60 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Kompetencje równoważne z zaliczeniem przedmiotów Wstęp do Informatyki i Matematyka Dyskretna.
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest przedstawienie podstawowych struktur danych i algorytmów oraz osiągnięcie zakładanych efektów kształcenia.

14.	<p>Treści programowe</p> <ol style="list-style-type: none"> 1. Przegląd metod projektowania efektywnych algorytmów: dziel i zwyciężaj, programowanie dynamiczne, metoda zachłanna. 2. Złożoność obliczeniowa algorytmu (pesymistyczna, oczekiwana, zamortyzowana). Przykłady analizy kosztu. 3. Sortowanie: Heapsort i Quicksort. Model drzew decyzyjnych i dolne ograniczenie na problem sortowania. Sortowanie w czasie liniowym: Countsort, Radixsort, Bucketsort. 4. Selekcja: algorytmy Hoarea i magicznych piątek. 5. Kolejki priorytetowe: kopce binarne, dwumianowe i Fibonacciego. Zastosowania w problemie najkrótszych ścieżek i minimalnego drzewa rozpinającego. 6. Scalanie. Drzewa turniejowe. Sortowanie zewnętrzne. 7. Wyszukiwanie i problem słownika. Drzewa wyszukiwań binarnych, zrownoważone drzewa wyszukiwań binarnych (AVL, 2-3-4-drzewa, drzewa czerwono-czarne). Optymalne drzewa wyszukiwań binarnych. Haszowanie. Wyszukiwanie pozycyjne. 8. Wyszukiwanie zewnętrzne - B-drzewa. 9. Problem sumowania zbiorów rozłącznych i jego zastosowania. 10. Algorytmy grafowe: przepływy w sieciach, skojarzenia. 11. Algorytmy na tekstach. Wyszukiwanie wzorca. Drzewa sufiksowe. 12. Geometria obliczeniowa. Lokalizacja punktu. Otoczka wypukła. Technika zamiatania. 13. Algorytmy algebraiczne i teoriolicebne. FFT. Szybkie mnożenie liczb i wielomianów. 14. NP-zupełność. Algorytmy aproksymacyjne dla problemów obliczeniowo trudnych. Heurystyki dla problemów trudnych (algorytmy genetyczne, simulated annealing). 15. Modele obliczeń równoległych: PRAM, tablica procesorów, hiperkostka. Algorytmy równoległe. Klasa NC i problemy P-zupełne. 16. Specjalne modele obliczeń: sieci komparatorów, obwody logiczne. 17. Algorytmy randomizacyjne. Przykłady w dziedzinach: struktury danych, geometria obliczeniowa, algorytmy grafowe, algorytmy równoległe. 																		
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">zna podstawowe struktury danych oraz ich właściwości</td> <td style="padding: 2px;">K_W06</td> </tr> <tr> <td style="padding: 2px;">zna i rozumie wybrane techniki i paradygmaty konstrukcji algorytmów</td> <td style="padding: 2px;">K_W06</td> </tr> <tr> <td style="padding: 2px;">posiada wiedzę na temat znanych rozwiązań dla typowych zagadnień algorytmicznych</td> <td style="padding: 2px;">K_W06</td> </tr> <tr> <td style="padding: 2px;">rozumie podstawowe pojęcia i techniki z zakresu analizy złożoności obliczeniowej algorytmów</td> <td style="padding: 2px;">K_W06</td> </tr> </table> <p>Umiejętności</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">potrafi zastosować poznane struktury danych do konstrukcji algorytmów dla wybranych zadań obliczeniowych</td> <td style="padding: 2px;">K_U06, Inż_U01</td> </tr> <tr> <td style="padding: 2px;">umie dowodzić poprawności konstruowanych algorytmów</td> <td style="padding: 2px;">K_U06</td> </tr> <tr> <td style="padding: 2px;">umie analizować złożoność obliczeniową (czasową i pamięciową) badanych algorytmów</td> <td style="padding: 2px;">K_U06</td> </tr> <tr> <td style="padding: 2px;">potrafi pisemnie przedstawić argumenty dotyczące poprawności i złożoności obliczeniowej wybranych rozwiązań algorytmicznych</td> <td style="padding: 2px;">K_U06</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">potrafi opowiadać o problemach algorytmicznych w sposób gwarantujący zrozumienie treści przez słuchaczy</td> <td style="padding: 2px;">K_K01, K_K02, K_K04</td> </tr> </table>	zna podstawowe struktury danych oraz ich właściwości	K_W06	zna i rozumie wybrane techniki i paradygmaty konstrukcji algorytmów	K_W06	posiada wiedzę na temat znanych rozwiązań dla typowych zagadnień algorytmicznych	K_W06	rozumie podstawowe pojęcia i techniki z zakresu analizy złożoności obliczeniowej algorytmów	K_W06	potrafi zastosować poznane struktury danych do konstrukcji algorytmów dla wybranych zadań obliczeniowych	K_U06, Inż_U01	umie dowodzić poprawności konstruowanych algorytmów	K_U06	umie analizować złożoność obliczeniową (czasową i pamięciową) badanych algorytmów	K_U06	potrafi pisemnie przedstawić argumenty dotyczące poprawności i złożoności obliczeniowej wybranych rozwiązań algorytmicznych	K_U06	potrafi opowiadać o problemach algorytmicznych w sposób gwarantujący zrozumienie treści przez słuchaczy	K_K01, K_K02, K_K04
zna podstawowe struktury danych oraz ich właściwości	K_W06																		
zna i rozumie wybrane techniki i paradygmaty konstrukcji algorytmów	K_W06																		
posiada wiedzę na temat znanych rozwiązań dla typowych zagadnień algorytmicznych	K_W06																		
rozumie podstawowe pojęcia i techniki z zakresu analizy złożoności obliczeniowej algorytmów	K_W06																		
potrafi zastosować poznane struktury danych do konstrukcji algorytmów dla wybranych zadań obliczeniowych	K_U06, Inż_U01																		
umie dowodzić poprawności konstruowanych algorytmów	K_U06																		
umie analizować złożoność obliczeniową (czasową i pamięciową) badanych algorytmów	K_U06																		
potrafi pisemnie przedstawić argumenty dotyczące poprawności i złożoności obliczeniowej wybranych rozwiązań algorytmicznych	K_U06																		
potrafi opowiadać o problemach algorytmicznych w sposób gwarantujący zrozumienie treści przez słuchaczy	K_K01, K_K02, K_K04																		
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Wprowadzenie do algorytmów, PWN, 2012. • Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman; Algorytmy i struktury danych; Helion. • Dexter Kozen, The Design and Analysis of Algorithms, Springer 1990. • Jon Kleinberg, Eva Tardos, Algorithm Design, Pearson 2006. • Notatki do wykładu. 																		
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>prezentacja rozwiązania zadania, napisanie programu komputerowego, egzamin pisemny</p>																		

18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Aby zaliczyć przedmiot, należy uzyskać zaliczenie pracowni i zaliczenie ćwiczeń oraz zdać egzamin. Do zaliczenia pracowni należy zdobyć wymaganą, podaną w regulaminie pracowni, liczbę punktów za napisanie programów rozwiązujących zadane problemy algorytmiczne. Liczba punktów za poszczególne programy zależy od ich poprawności i efektywności. Zależność ta jest szczegółowo opisana w regulaminie pracowni.</p> <p>Do zaliczenia ćwiczeń należy zdobyć wymaganą, podaną w regulaminie ćwiczeń, liczbę punktów, uzyskanych za deklaracje rozwiązań oraz ich prezentacje.</p> <p>Egzamin ma formę pisemną i składa się z dwóch części. Podczas pierwszej części, trwającej 2,5 godziny, studenci otrzymują 20 pytań lub bardzo prostych zadań, mających na celu sprawdzenie stopnia opanowania wiedzy. W drugiej części, trwającej 4,5 godziny, studenci otrzymują sześć zadań, spośród których powinni wybrać i rozwiązać trzy dowolne. Te zadania sprawdzają stopień opanowania umiejętności określone w zakładanych efektach uczenia się.</p> <p>Do zdania egzaminu należy uzyskać wymagane, określone w regulaminie przedmiotu, liczby punktów za poszczególne części egzaminu.</p>																		
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="220 696 1428 763"> <tr> <td>wykład</td> <td>60 godz.</td> </tr> <tr> <td>ćwiczenio-pracownia</td> <td>60 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="220 842 1428 999"> <tr> <td>studiowanie notatek i literatury</td> <td>15 godz.</td> </tr> <tr> <td>rozwiązywanie zadań z list</td> <td>60 godz.</td> </tr> <tr> <td>udział w egzaminie</td> <td>7 godz.</td> </tr> <tr> <td>przygotowanie do egzaminu</td> <td>35 godz.</td> </tr> <tr> <td>rozwiązywanie zadań programistycznych</td> <td>60 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="220 1084 1428 1151"> <tr> <td>Łączna liczba godzin</td> <td>297 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>12</td> </tr> </table>	wykład	60 godz.	ćwiczenio-pracownia	60 godz.	studiowanie notatek i literatury	15 godz.	rozwiązywanie zadań z list	60 godz.	udział w egzaminie	7 godz.	przygotowanie do egzaminu	35 godz.	rozwiązywanie zadań programistycznych	60 godz.	Łączna liczba godzin	297 godz.	Liczba punktów ECTS	12
wykład	60 godz.																		
ćwiczenio-pracownia	60 godz.																		
studiowanie notatek i literatury	15 godz.																		
rozwiązywanie zadań z list	60 godz.																		
udział w egzaminie	7 godz.																		
przygotowanie do egzaminu	35 godz.																		
rozwiązywanie zadań programistycznych	60 godz.																		
Łączna liczba godzin	297 godz.																		
Liczba punktów ECTS	12																		

2 Przedmioty informatyczne (I1)

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Architektury systemów komputerowych Computer Architecture for Programmers
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DP6ASK
6.	Rodzaj przedmiotu/modułu Informatyczny podstawowy (I1) ze znacznikiem ASK
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, pracownia — 15 godzin, ćwiczenie-pracownia — 30 godzin, repetytorium — 15 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty <ul style="list-style-type: none">• Kurs: Wstęp do programowania w języku C• Logika cyfrowa (opcjonalnie) Niezbędne kompetencje <ul style="list-style-type: none">• Umiejętność obsługi wiersza poleceń systemu uniksowego• Dobra znajomość języka C
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest wprowadzenie słuchaczy w tematykę budowy i zasady działania komputerów z punktu widzenia programisty. Po zakończeniu przedmiotu studenci będą rozumieć: reprezentację programów i danych w pamięci, wpływ architektury komputera na wydajność programów oraz jak pisać kod wolny od podatności na ataki.

14.	<p>Treści programowe</p> <ol style="list-style-type: none"> 1. Reprezentacja danych w pamięci komputera i reprezentacja pośrednia programów: kod trójadre-sowy, graf przepływu sterowania. 2. Dziwne właściwości arytmetyki całkowitoliczbowej i zmiennopozycyjnej (standard IEEE754). 3. Składnia assemblera: mnemoniki, operandy, dyrektywy. 4. Translacja konstrukcji języka C do assemblera (wyrażenia, instrukcje sterujące, typy rekordowe i tablicowe) 5. Konwencja wołania procedur i organizacja stosu wywołań (ABI) 6. Konsolidacja i ładowanie: ELF, sekcje, relokacje, układ pamięci, biblioteki i konsolidator dyna-miczny (ld.so). 7. Ataki na oprogramowanie: błąd przepełnienia bufora, wstrzykiwanie kodu, technika ROP, tech-niki niwelowania podatności (ASLR, WX). 8. Organizacja pamięci operacyjnej i masowej (magnetycznej i półprzewodnikowej) - czas dostępu. 9. Komunikacja procesora z urządzeniami peryferyjnymi: przerwania, MMIO, DMA. 10. Pamięć podręczna: zasada lokalności dostępu, organizacja pamięci podręcznej, polityki wymiany i zapisu. 11. Analiza i optymalizacja programów pod kątem lokalności dostępu do pamięci. 12. Pamięć wirtualna: motywacja, przestrzeń adresowa, zbiór roboczy i rezydentny, błąd strony, stronicowanie na żądanie. 13. Translacja adresów: deskryptory stron, hierarchiczna tablica stron, TLB, przełączanie przestrzeni adresowych. 14. Mikroarchitektura: przetwarzanie superskalarne, OoO, spekulacja, przewidywanie skoków. 																								
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 913 1423 1070"> <tr> <td>Wie jak programy komunikują się ze sprzętem i systemem operacyjnym</td> <td>K_W07, Inż_W01</td> </tr> <tr> <td>Zna wpływ architektury maszyny na implementację kompilatorów i systemów operacyjnych</td> <td>K_W07, Inż_W01</td> </tr> <tr> <td>Wie jak wykorzystać organizację maszyny do implementacji wydajnych pro-gramów</td> <td>K_W07, Inż_W01</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 1160 1423 1664"> <tr> <td>Umie pisać programy zachowujące się poprawnie w obliczu ograniczeń maszy-nowej reprezentacji typów numerycznych</td> <td>K_U03, Inż_U05</td> </tr> <tr> <td>Potrafi wybrać i zastosować właściwe narzędzia i metodologie do mierzenia wydajności kodu programów</td> <td>K_U09, Inż_U02</td> </tr> <tr> <td>Potrafi przeprowadzić analizę kodu pod względem wykorzystania pamięci pod-ręcznej i przeorganizować go celem optymalizacji</td> <td>K_U09, Inż_U01, Inż_U04</td> </tr> <tr> <td>Umie przeanalizować kod maszynowy generowany przez kompilator pod wzglę-dem poprawności i wydajności</td> <td>K_U09, Inż_U02, Inż_U04</td> </tr> <tr> <td>Potrafi użyć narzędzi profilujących do znalezienia fragmentów programu wy-magających optymalizacji</td> <td>K_U07, Inż_U01, Inż_U04</td> </tr> <tr> <td>Umie przeanalizować kod programu pod kątem występowania podatności na ataki</td> <td>K_U09, Inż_U02</td> </tr> <tr> <td>Potrafi wykonać inżynierię wsteczną fragmentów kodu maszynowego celem ustalenia pełnionych przez nie funkcji</td> <td>K_U09, Inż_U02, Inż_U04</td> </tr> <tr> <td>Pomaga członkom zespołu w pisaniu oprogramowania wolnego od usterek zwią-zanych z niskopoziomymi aspektami działania komputerów</td> <td>K_U14</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 1753 1423 1816"> <tr> <td>Potrafi właściwie wskazać te części projektu komercyjnego, które oplaca się optymalizować</td> <td>K_K02, K_K04</td> </tr> </table>	Wie jak programy komunikują się ze sprzętem i systemem operacyjnym	K_W07, Inż_W01	Zna wpływ architektury maszyny na implementację kompilatorów i systemów operacyjnych	K_W07, Inż_W01	Wie jak wykorzystać organizację maszyny do implementacji wydajnych pro-gramów	K_W07, Inż_W01	Umie pisać programy zachowujące się poprawnie w obliczu ograniczeń maszy-nowej reprezentacji typów numerycznych	K_U03, Inż_U05	Potrafi wybrać i zastosować właściwe narzędzia i metodologie do mierzenia wydajności kodu programów	K_U09, Inż_U02	Potrafi przeprowadzić analizę kodu pod względem wykorzystania pamięci pod-ręcznej i przeorganizować go celem optymalizacji	K_U09, Inż_U01, Inż_U04	Umie przeanalizować kod maszynowy generowany przez kompilator pod wzglę-dem poprawności i wydajności	K_U09, Inż_U02, Inż_U04	Potrafi użyć narzędzi profilujących do znalezienia fragmentów programu wy-magających optymalizacji	K_U07, Inż_U01, Inż_U04	Umie przeanalizować kod programu pod kątem występowania podatności na ataki	K_U09, Inż_U02	Potrafi wykonać inżynierię wsteczną fragmentów kodu maszynowego celem ustalenia pełnionych przez nie funkcji	K_U09, Inż_U02, Inż_U04	Pomaga członkom zespołu w pisaniu oprogramowania wolnego od usterek zwią-zanych z niskopoziomymi aspektami działania komputerów	K_U14	Potrafi właściwie wskazać te części projektu komercyjnego, które oplaca się optymalizować	K_K02, K_K04
Wie jak programy komunikują się ze sprzętem i systemem operacyjnym	K_W07, Inż_W01																								
Zna wpływ architektury maszyny na implementację kompilatorów i systemów operacyjnych	K_W07, Inż_W01																								
Wie jak wykorzystać organizację maszyny do implementacji wydajnych pro-gramów	K_W07, Inż_W01																								
Umie pisać programy zachowujące się poprawnie w obliczu ograniczeń maszy-nowej reprezentacji typów numerycznych	K_U03, Inż_U05																								
Potrafi wybrać i zastosować właściwe narzędzia i metodologie do mierzenia wydajności kodu programów	K_U09, Inż_U02																								
Potrafi przeprowadzić analizę kodu pod względem wykorzystania pamięci pod-ręcznej i przeorganizować go celem optymalizacji	K_U09, Inż_U01, Inż_U04																								
Umie przeanalizować kod maszynowy generowany przez kompilator pod wzglę-dem poprawności i wydajności	K_U09, Inż_U02, Inż_U04																								
Potrafi użyć narzędzi profilujących do znalezienia fragmentów programu wy-magających optymalizacji	K_U07, Inż_U01, Inż_U04																								
Umie przeanalizować kod programu pod kątem występowania podatności na ataki	K_U09, Inż_U02																								
Potrafi wykonać inżynierię wsteczną fragmentów kodu maszynowego celem ustalenia pełnionych przez nie funkcji	K_U09, Inż_U02, Inż_U04																								
Pomaga członkom zespołu w pisaniu oprogramowania wolnego od usterek zwią-zanych z niskopoziomymi aspektami działania komputerów	K_U14																								
Potrafi właściwie wskazać te części projektu komercyjnego, które oplaca się optymalizować	K_K02, K_K04																								
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • Computer Systems: A Programmer's Perspective (3rd edition) Randal E. Bryant, David R. O'Hallaron 																								
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>Egzamin pisemny, prezentacja rozwiązań zadań przy tablicy, napisanie i prezentacja niedużych progra-mów, znalezienie luki w podatnym oprogramowaniu, analiza zachowania programów.</p>																								

18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Do zaliczenia ćwiczeń i pracowni należy zdobyć co najmniej połowę liczby punktów za zadania ćwiczeniowe i programistyczne. Punkty za wszystkie wyżej wymienione aktywności liczą się łącznie. Na punkty z poszczególnych aktywności mogą być narzucone minimalne progi zaliczeniowe.</p> <p>Egzamin końcowy jest w formie pisemnej i wymaga zdobycia co najmniej połowy punktów. Zadania mogą mieć następujące formy: analiza kodu programu, prezentacja działania mechanizmu sprzętowego, pisanie niewielkich fragmentów kodu, test wielokrotnego wyboru.</p>																				
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 443 1428 568"> <tr> <td>repetitorium</td> <td>15 godz.</td> </tr> <tr> <td>ćwiczenie-pracownia</td> <td>30 godz.</td> </tr> <tr> <td>pracownia</td> <td>15 godz.</td> </tr> <tr> <td>wykład</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 647 1428 772"> <tr> <td>rozwiązywanie zadań programistycznych</td> <td>50 godz.</td> </tr> <tr> <td>udział w egzaminie</td> <td>3 godz.</td> </tr> <tr> <td>przygotowanie do egzaminu</td> <td>10 godz.</td> </tr> <tr> <td>rozwiązywanie zadań z list</td> <td>60 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 860 1428 920"> <tr> <td>Łączna liczba godzin</td> <td>213 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>8</td> </tr> </table>	repetitorium	15 godz.	ćwiczenie-pracownia	30 godz.	pracownia	15 godz.	wykład	30 godz.	rozwiązywanie zadań programistycznych	50 godz.	udział w egzaminie	3 godz.	przygotowanie do egzaminu	10 godz.	rozwiązywanie zadań z list	60 godz.	Łączna liczba godzin	213 godz.	Liczba punktów ECTS	8
repetitorium	15 godz.																				
ćwiczenie-pracownia	30 godz.																				
pracownia	15 godz.																				
wykład	30 godz.																				
rozwiązywanie zadań programistycznych	50 godz.																				
udział w egzaminie	3 godz.																				
przygotowanie do egzaminu	10 godz.																				
rozwiązywanie zadań z list	60 godz.																				
Łączna liczba godzin	213 godz.																				
Liczba punktów ECTS	8																				

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Bazy danych Introduction to databases
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DP6BD
6.	Rodzaj przedmiotu/modułu Informatyczny podstawowy (I1) ze znacznikiem BD
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenio-pracowni – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Niezbędne kompetencje: <ul style="list-style-type: none"> • Podstawy logiki i teorii mnogości (pojęcie relacji i rachunku zbiorów, formuły z kwantyfikatorami). • Podstawy programowania (znajomość podstawowych cech systemów informatycznych). • Umiejętność programowania w wybranym języku w stopniu wystarczającym do napisania aplikacji do projektu bazodanowego.
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest zapoznanie słuchaczy z typowymi systemami baz danych, głównie z systemami relacyjnymi, oraz opanowanie kluczowych umiejętności związanych z użytkowaniem i tworzeniem baz danych oraz stosowaniem systemów bazodanowych.

14.	<p>Treści programowe</p> <p>Program wykładu</p> <ol style="list-style-type: none"> 1. Model relacyjny. Podstawowe pojęcia (relacja) oraz języki zapytań (algebra relacji, rachunki relacyjne). 2. Teoria relacyjnych baz danych (zależności funkcyjne, postaci normalne, denormalizacja). 3. SQL jako standard relacyjnego języka zapytań i opisu danych. Zapytania SQL, elementy relacyjnej bazy danych (tabele, dziedziny, perspektywy, wyzwalacze i asercje, funkcje, użytkownicy i uprawnienia). 4. SQL w aplikacjach bazodanowych: osadzony SQL, programistyczny SQL, PLSQL. 5. Modelowanie konceptualne baz danych (diagramy E-R, UML). 6. Tworzenie bazy danych w oparciu o model konceptualny. 7. Elementy systemów zarządzania bazami danych (transakcje, optymalizacja, bezpieczeństwo danych). 8. Inne modele baz danych (m.in. grafowe bazy danych). <p>Program ćwiczeń i pracowni</p> <ol style="list-style-type: none"> 1. Zajęcia konwersatoryjne (ćwiczenia) pomagające w praktycznym opanowaniu pojęć z teorii bazy danych, zasad działania systemów zarządzania bazami danych oraz projektowania relacyjnych baz danych. 2. Zajęcia laboratoryjne (pracownia) pozwalające zapoznać się z profesjonalnymi systemami zarządzania bazami danych (PostgreSQL itp.) i opanować język SQL. 3. Wykonanie kompletnego projektu niewielkiej bazy danych wraz z aplikacją użytkownika. 																																
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 952 1426 1205"> <tr> <td>zna pojęcia modelu relacyjnego oraz zasady dobrego modelowania relacyjnych baz danych</td> <td>K_W07, Inż_W01</td> </tr> <tr> <td>jest świadomy istnienia innych modeli, ich zalet i wad w porównaniu do modelu relacyjnego</td> <td>K_W07, Inż_W01</td> </tr> <tr> <td>zna język SQL w zakresie opisu bazy danych i wyszukiwania informacji w bazie danych</td> <td>K_W07, Inż_W01</td> </tr> <tr> <td>zna zasady działania systemów zarządzania bazami danych w zakresie optymalizacji, wielodostępu i bezpieczeństwa</td> <td>K_W07, Inż_W01</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 1294 1426 1771"> <tr> <td>projektuje modele relacyjnej bazy danych i formułuje zapytania w algebrze relacji i rachunkach relacyjnych</td> <td>K_U07, Inż_U04</td> </tr> <tr> <td>definiuje bazę danych w SQL za użyciem dostępnych i właściwie dobranych elementów i struktur tego języka</td> <td>K_U07, Inż_U04</td> </tr> <tr> <td>formułuje efektywne i poprawne zapytania SQL</td> <td>K_U03, Inż_U4</td> </tr> <tr> <td>projektuje bazy danych dostosowane do potrzeb użytkownika i spełniające wymogi dobrego i efektywnego projektu</td> <td>K_U08, Inż_U04</td> </tr> <tr> <td>tworzy bazy danych w oparciu o podany projekt</td> <td>K_U08, Inż_U05</td> </tr> <tr> <td>tworzy aplikacje bazodanowe zgodnie z zadaną specyfikacją</td> <td>K_U08, Inż_U05</td> </tr> <tr> <td>potrafi zastosować posiadaną wiedzę do realizacji efektywnych rozwiązań w bazach danych</td> <td>K_U09, Inż_U04</td> </tr> <tr> <td>potrafi jasno i zrozumiale przedstawiać własne pomysły (np. rozwiązanie zadania, element projektu bazy danych, fragment kodu), a także prosić o wyjaśnienia w przypadku problemów ze zrozumieniem pomysłów i rozwiązań proponowanych przez innych.</td> <td>K_U12</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 1854 1426 2112"> <tr> <td>Jest świadomy ograniczeń własnej wiedzy i umiejętności, adekwatnie ocenia poziom swoich kompetencji, zna swoje mocne i słabe strony, rozumie konieczność doskonalenia swoich zawodowych kompetencji</td> <td>K_K01</td> </tr> <tr> <td>Jest świadom roli i znaczenia wiedzy w rozwiązywaniu problemów o charakterze poznawczym i praktycznym</td> <td>K_K02</td> </tr> <tr> <td>Aktywnie prezentuje krytyczną postawę wobec stwierdzeń, uwag i wniosków</td> <td>K_K03, K_K04</td> </tr> <tr> <td>Wykazuje samodzielność myślenia i działania przy rozwiązywaniu problemów i wykonywaniu zadań w zakresie projektowania i tworzenia baz danych</td> <td>K_K05</td> </tr> </table>	zna pojęcia modelu relacyjnego oraz zasady dobrego modelowania relacyjnych baz danych	K_W07, Inż_W01	jest świadomy istnienia innych modeli, ich zalet i wad w porównaniu do modelu relacyjnego	K_W07, Inż_W01	zna język SQL w zakresie opisu bazy danych i wyszukiwania informacji w bazie danych	K_W07, Inż_W01	zna zasady działania systemów zarządzania bazami danych w zakresie optymalizacji, wielodostępu i bezpieczeństwa	K_W07, Inż_W01	projektuje modele relacyjnej bazy danych i formułuje zapytania w algebrze relacji i rachunkach relacyjnych	K_U07, Inż_U04	definiuje bazę danych w SQL za użyciem dostępnych i właściwie dobranych elementów i struktur tego języka	K_U07, Inż_U04	formułuje efektywne i poprawne zapytania SQL	K_U03, Inż_U4	projektuje bazy danych dostosowane do potrzeb użytkownika i spełniające wymogi dobrego i efektywnego projektu	K_U08, Inż_U04	tworzy bazy danych w oparciu o podany projekt	K_U08, Inż_U05	tworzy aplikacje bazodanowe zgodnie z zadaną specyfikacją	K_U08, Inż_U05	potrafi zastosować posiadaną wiedzę do realizacji efektywnych rozwiązań w bazach danych	K_U09, Inż_U04	potrafi jasno i zrozumiale przedstawiać własne pomysły (np. rozwiązanie zadania, element projektu bazy danych, fragment kodu), a także prosić o wyjaśnienia w przypadku problemów ze zrozumieniem pomysłów i rozwiązań proponowanych przez innych.	K_U12	Jest świadomy ograniczeń własnej wiedzy i umiejętności, adekwatnie ocenia poziom swoich kompetencji, zna swoje mocne i słabe strony, rozumie konieczność doskonalenia swoich zawodowych kompetencji	K_K01	Jest świadom roli i znaczenia wiedzy w rozwiązywaniu problemów o charakterze poznawczym i praktycznym	K_K02	Aktywnie prezentuje krytyczną postawę wobec stwierdzeń, uwag i wniosków	K_K03, K_K04	Wykazuje samodzielność myślenia i działania przy rozwiązywaniu problemów i wykonywaniu zadań w zakresie projektowania i tworzenia baz danych	K_K05
zna pojęcia modelu relacyjnego oraz zasady dobrego modelowania relacyjnych baz danych	K_W07, Inż_W01																																
jest świadomy istnienia innych modeli, ich zalet i wad w porównaniu do modelu relacyjnego	K_W07, Inż_W01																																
zna język SQL w zakresie opisu bazy danych i wyszukiwania informacji w bazie danych	K_W07, Inż_W01																																
zna zasady działania systemów zarządzania bazami danych w zakresie optymalizacji, wielodostępu i bezpieczeństwa	K_W07, Inż_W01																																
projektuje modele relacyjnej bazy danych i formułuje zapytania w algebrze relacji i rachunkach relacyjnych	K_U07, Inż_U04																																
definiuje bazę danych w SQL za użyciem dostępnych i właściwie dobranych elementów i struktur tego języka	K_U07, Inż_U04																																
formułuje efektywne i poprawne zapytania SQL	K_U03, Inż_U4																																
projektuje bazy danych dostosowane do potrzeb użytkownika i spełniające wymogi dobrego i efektywnego projektu	K_U08, Inż_U04																																
tworzy bazy danych w oparciu o podany projekt	K_U08, Inż_U05																																
tworzy aplikacje bazodanowe zgodnie z zadaną specyfikacją	K_U08, Inż_U05																																
potrafi zastosować posiadaną wiedzę do realizacji efektywnych rozwiązań w bazach danych	K_U09, Inż_U04																																
potrafi jasno i zrozumiale przedstawiać własne pomysły (np. rozwiązanie zadania, element projektu bazy danych, fragment kodu), a także prosić o wyjaśnienia w przypadku problemów ze zrozumieniem pomysłów i rozwiązań proponowanych przez innych.	K_U12																																
Jest świadomy ograniczeń własnej wiedzy i umiejętności, adekwatnie ocenia poziom swoich kompetencji, zna swoje mocne i słabe strony, rozumie konieczność doskonalenia swoich zawodowych kompetencji	K_K01																																
Jest świadom roli i znaczenia wiedzy w rozwiązywaniu problemów o charakterze poznawczym i praktycznym	K_K02																																
Aktywnie prezentuje krytyczną postawę wobec stwierdzeń, uwag i wniosków	K_K03, K_K04																																
Wykazuje samodzielność myślenia i działania przy rozwiązywaniu problemów i wykonywaniu zadań w zakresie projektowania i tworzenia baz danych	K_K05																																

16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • Jeffrey D. Ullman, Jennifer Widom, Podstawowy Kurs Systemów Baz Danych, WNT, Warszawa 1999; • Garcia-Molina H., Ullman J.D., Widom J., Implementacja systemów baz danych, WNT, 2003 (seria: Klasyka Informatyki); • Garcia-Molina H., Ullman J.D., Widom J., Database Systems: The Complete Book (suma dwóch powyższych pozycji); • Thomas Connolly, Carolyn Begg, Database Systems, Addison Wesley 2002, także po polsku: ReadMe 2004; Date C. J., An Introduction to Database Systems, vol. II, Addison-Wesley Pub. Comp., również WNT – W-wa, (seria: Klasyka Informatyki), 2000; • R. Ramakrishnan, J. Gehrke, Database Management Systems, McGraw-Hill Education, 2003 • dokumentacja systemu PostgreSQL (http://www.postgresql.org/docs/) 																						
17.	<p>Metody weryfikacji zakładanych efektów uczenia się egzamin, prezentacja projektu, praktyczne sprawdziany SQL</p>																						
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Zaliczenie ćwiczeń i pracowni na podstawie liczby punktów zdobytych za poniższe aktywności:</p> <ul style="list-style-type: none"> • prezentacja i dyskusja rozwiązań zadań z listy opublikowanej przed zajęciami; • sprawdziany pisemne ze znajomości modelu relacyjnego i relacyjnych języków zapytań; • sprawdziany praktyczne ze znajomości języka SQL (tworzenie zapytań pod kontrolą nauczyciela w oparciu o udostępnioną dokumentację); • zaliczenie projektu bazy danych oraz prezentacja aplikacji; • egzamin pisemny ze znajomości modelu relacyjnego oraz podstaw działania systemów zarządzania bazami danych; 																						
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="223 996 1428 1064"> <tr> <td>ćwiczenie-pracownia</td> <td>30 godz.</td> </tr> <tr> <td>wykład</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="223 1142 1428 1366"> <tr> <td>przygotowanie raportu/prezentacji</td> <td>2 godz.</td> </tr> <tr> <td>przygotowanie do sprawdzianów/kolokwii</td> <td>25 godz.</td> </tr> <tr> <td>przygotowanie do ćwiczeń</td> <td>10 godz.</td> </tr> <tr> <td>studiowanie tematyki wykładów i literatury</td> <td>20 godz.</td> </tr> <tr> <td>przygotowanie do pracowni</td> <td>20 godz.</td> </tr> <tr> <td>praca nad projektem</td> <td>30 godz.</td> </tr> <tr> <td>przygotowanie do egzaminu</td> <td>5 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="223 1444 1428 1512"> <tr> <td>Łączna liczba godzin</td> <td>172 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>6</td> </tr> </table>	ćwiczenie-pracownia	30 godz.	wykład	30 godz.	przygotowanie raportu/prezentacji	2 godz.	przygotowanie do sprawdzianów/kolokwii	25 godz.	przygotowanie do ćwiczeń	10 godz.	studiowanie tematyki wykładów i literatury	20 godz.	przygotowanie do pracowni	20 godz.	praca nad projektem	30 godz.	przygotowanie do egzaminu	5 godz.	Łączna liczba godzin	172 godz.	Liczba punktów ECTS	6
ćwiczenie-pracownia	30 godz.																						
wykład	30 godz.																						
przygotowanie raportu/prezentacji	2 godz.																						
przygotowanie do sprawdzianów/kolokwii	25 godz.																						
przygotowanie do ćwiczeń	10 godz.																						
studiowanie tematyki wykładów i literatury	20 godz.																						
przygotowanie do pracowni	20 godz.																						
praca nad projektem	30 godz.																						
przygotowanie do egzaminu	5 godz.																						
Łączna liczba godzin	172 godz.																						
Liczba punktów ECTS	6																						

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Introduction to Linear Optimization Introduction to Linear Optimization
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DI1ILinOpt
6.	Rodzaj przedmiotu/modułu Informatyczny 1
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów ((jeśli obowiązuje)) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład – 30 godzin, ćwiczenie-pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: <ul style="list-style-type: none">• Algebra Niezbędne kompetencje: <ul style="list-style-type: none">• Basic programming skills
13.	Cele przedmiotu This course offers a thorough introduction to the theory of linear optimization, with a particular focus on both the mathematical background of the area as well as empowering the student to be able to model many computer science problems as linear systems, and solve such systems on a computer with the use of a linear programming solver.
14.	Treści programowe <ul style="list-style-type: none">• Modeling with linear and integer programs.• Convexity and its properties.• Hyperplanes, polytopes and polyhedra. Farkas lemma.• Duality in linear optimization.• Ellipsoid method: an algorithm for solving linear programs.• Solution methods for large LPs: separation oracles and column generation methods.• Network flows and their algorithms. Duality between network flows and graph cuts.• Integrality of selected combinatorial polytopes. Linear relaxations of integer programs.

15.	Zakładane efekty uczenia się	
	Wiedza	
	Knows the basic theory of Linear Programming	K_W08
	Knows the concept of LP duality	K_W08
	Knows basic algorithms to solve Linear Programms	K_W08
	Wiedza	
	Is capable of modeling a combinatorial optimization problem with Integer Programming	K_U10
	Can use an LP solver to solve simple Linear Programms	K_U10
	Can identify separation oracles for selected polyhedra	K_U10
16.	Literatura obowiązkowa i zalecana <ul style="list-style-type: none"> • Jiří Matoušek, Bernd Gärtner. Understanding and Using Linear Programming. Springer 2007 (https://link.springer.com/book/10.1007/978-3-540-30717-4) • Lecture notes of Alexander Shrijver (in particular, Section 2 of https://homepages.cwi.nl/~lex/files/dict.pdf) 	
17.	Metody weryfikacji zakładanych efektów kształcenia <ul style="list-style-type: none"> • Exam • Exercise solving • Writing small computer programms (solving basic LPs with LP solvers) 	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu <ul style="list-style-type: none"> • Collecting enough points for solving exercises and implementation tasks • Passing a final exam 	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	30 godz.
	ćwiczenie-pracownia	30 godz.
	Praca własna studenta	
	Preparing for the exam	15 godz.
	Solving theoretical exercises	30 godz.
	Programming tasks:	15 godz.
	Sumarycznie	
	Łączna liczba godzin	120 godz.
	Liczba punktów ECTS	6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Inżynieria oprogramowania Software engineering
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DP6IO
6.	Rodzaj przedmiotu/modułu Informatyczny podstawowy (I1) ze znacznikiem IO
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenio-pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Gotowość do pracy zespołowej
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest zdobycie podstawowych: praktycznej wiedzy i umiejętności wymaganych w pracy na stanowisku inżynier oprogramowania. Wykład ma na celu przekazanie studentom podstawowej, praktycznej wiedzy z zakresu inżynierii oprogramowania. Przedstawia informacje dotyczące wykorzystania dziedzin wiedzy inżynierii. Istnieje możliwość wygłoszenia paru wykładów przez specjalistów z czołowych zagranicznych i polskich firm np: VOLVO IT, Capgemini. Podczas ćwiczeń/pracowni studenci zespołowo realizują, zgodnie z najlepszymi praktykami 10 podstawowych dyscyplin inżynierii oprogramowania, projekt konkretnego systemu informatycznego. Projekt zawiera również fragmenty kodu. Podczas zajęć laboratoryjnych wytwarzane są diagramy dot. fragmentów projektowanego systemu oraz fragmenty kodu. Realizowany projekt ma za zadanie przygotowanie wg standardów światowych do pracy w przemyśle wytwarzania oprogramowania.

14.	<p>Treści programowe</p> <ul style="list-style-type: none"> • Wprowadzenie - historia inżynierii oprogramowania – kodeks etyki informatyka (2g.) • Omówienie dyscyplin podstawowych i pokrewnych inżynierii oprogramowania (2g.) • Studium wykonalności (2g.) • Zarządzanie wymaganiami (2g.) • Projektowanie oprogramowania (2g.) • Kodowanie (2g.) • Testowanie oprogramowania (2g.) • Jakość oprogramowania (2g.) • Eksploatacja oprogramowania (2g.) • Zarządzanie konfiguracją oprogramowania (2g.), • Procesy w życiu oprogramowania (2g.), • Zarządzanie zespołowym wytwarzaniem oprogramowania (2g.), • Narzędzia do zarządzania konfiguracjami i wersjami oprogramowania (2g.). • Kolokwium (4g.) 																						
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%; padding: 5px;">Zna metody inżynierii oprogramowania dotyczące analizy i specyfikacji problemu informatycznego (wzorce projektowe, architektura oprogramowania, analiza i projektowanie obiektowe).</td> <td style="width: 20%; padding: 5px;">K_W07, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">Zna metody i narzędzia wytwarzania i rozwijania oprogramowania (narzędzia do analizy wymagań i modelowania, narzędzia do testowania, narzędzia do podglądu kodu, narzędzia do zarządzania projektem konfiguracjami i wersjami oprogramowania).</td> <td style="padding: 5px;">K_W07, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">Ma wiedzę na temat cyklu życia projektu informatycznego, walidacji i weryfikacji, technik utrzymywania oprogramowania (refaktoryzacji) i potrafi ją stosować w przypadku podstawowych systemów informatycznych.</td> <td style="padding: 5px;">K_W07, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">Ma podstawową wiedzę dotyczącą prawnych, społecznych i etycznych aspektów pracy informatyka, ze szczególnym uwzględnieniem zagadnień inżynierii oprogramowania.</td> <td style="padding: 5px;">K_W07, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">Ma podstawową wiedzę o strukturze, działaniu i cyklach życia systemów informatycznych oraz o oprogramowaniu urządzeń komputerowych.</td> <td style="padding: 5px;">K_W07, Inż_W01</td> </tr> </table> <p>Umiejętności</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%; padding: 5px;">Potrafi projektować pisać, uruchamiać i testować programy w wybranych paradygmatach i środowiskach programistycznych.</td> <td style="width: 20%; padding: 5px;">K_U03, Inż_U01</td> </tr> <tr> <td style="padding: 5px;">Projektuje oprogramowanie zgodnie z metodyką obiektową, posługuje się wzorcami projektowymi i potrafi stworzyć model obiektowy prostego systemu w języku UML.</td> <td style="padding: 5px;">K_U08, Inż_U05</td> </tr> <tr> <td style="padding: 5px;">Potrafi wykonać prostą analizę funkcjonowania systemu informatycznego, stworzyć i przeprowadzić plan testowania, wziąć udział w inspekcji oprogramowania.</td> <td style="padding: 5px;">K_U08, Inż_U04</td> </tr> <tr> <td style="padding: 5px;">Przy formułowaniu i rozwiązywaniu zadań inżynierskich dostrzega ich aspekty systemowe i pozatechniczne, w szczególności potrafi dokonać wstępnej analizy ekonomicznej podejmowanych działań inżynierskich, w tym szacowania czasochłonności implementowania wybranych metod działania.</td> <td style="padding: 5px;">K_U07, Inż_U03</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%; padding: 5px;">Jest świadom roli i znaczenia wiedzy w rozwiązywaniu problemów o charakterze poznawczym i praktycznym typowych dla inżynierii oprogramowania.</td> <td style="width: 20%; padding: 5px;">K_K02</td> </tr> <tr> <td style="padding: 5px;">Wykazuje gotowość do pełnienia, w sposób odpowiedzialny i respektujący zasady etyki zawodowej, ról zawodowych wymagających kompetencji zdobywanych w ramach studiów na kierunku informatyka, w szczególności jest gotów do formułowania opinii o zagadnieniach związanych z inżynierią oprogramowania.</td> <td style="padding: 5px;">K_K04</td> </tr> </table>	Zna metody inżynierii oprogramowania dotyczące analizy i specyfikacji problemu informatycznego (wzorce projektowe, architektura oprogramowania, analiza i projektowanie obiektowe).	K_W07, Inż_W01	Zna metody i narzędzia wytwarzania i rozwijania oprogramowania (narzędzia do analizy wymagań i modelowania, narzędzia do testowania, narzędzia do podglądu kodu, narzędzia do zarządzania projektem konfiguracjami i wersjami oprogramowania).	K_W07, Inż_W01	Ma wiedzę na temat cyklu życia projektu informatycznego, walidacji i weryfikacji, technik utrzymywania oprogramowania (refaktoryzacji) i potrafi ją stosować w przypadku podstawowych systemów informatycznych.	K_W07, Inż_W01	Ma podstawową wiedzę dotyczącą prawnych, społecznych i etycznych aspektów pracy informatyka, ze szczególnym uwzględnieniem zagadnień inżynierii oprogramowania.	K_W07, Inż_W01	Ma podstawową wiedzę o strukturze, działaniu i cyklach życia systemów informatycznych oraz o oprogramowaniu urządzeń komputerowych.	K_W07, Inż_W01	Potrafi projektować pisać, uruchamiać i testować programy w wybranych paradygmatach i środowiskach programistycznych.	K_U03, Inż_U01	Projektuje oprogramowanie zgodnie z metodyką obiektową, posługuje się wzorcami projektowymi i potrafi stworzyć model obiektowy prostego systemu w języku UML.	K_U08, Inż_U05	Potrafi wykonać prostą analizę funkcjonowania systemu informatycznego, stworzyć i przeprowadzić plan testowania, wziąć udział w inspekcji oprogramowania.	K_U08, Inż_U04	Przy formułowaniu i rozwiązywaniu zadań inżynierskich dostrzega ich aspekty systemowe i pozatechniczne, w szczególności potrafi dokonać wstępnej analizy ekonomicznej podejmowanych działań inżynierskich, w tym szacowania czasochłonności implementowania wybranych metod działania.	K_U07, Inż_U03	Jest świadom roli i znaczenia wiedzy w rozwiązywaniu problemów o charakterze poznawczym i praktycznym typowych dla inżynierii oprogramowania.	K_K02	Wykazuje gotowość do pełnienia, w sposób odpowiedzialny i respektujący zasady etyki zawodowej, ról zawodowych wymagających kompetencji zdobywanych w ramach studiów na kierunku informatyka, w szczególności jest gotów do formułowania opinii o zagadnieniach związanych z inżynierią oprogramowania.	K_K04
Zna metody inżynierii oprogramowania dotyczące analizy i specyfikacji problemu informatycznego (wzorce projektowe, architektura oprogramowania, analiza i projektowanie obiektowe).	K_W07, Inż_W01																						
Zna metody i narzędzia wytwarzania i rozwijania oprogramowania (narzędzia do analizy wymagań i modelowania, narzędzia do testowania, narzędzia do podglądu kodu, narzędzia do zarządzania projektem konfiguracjami i wersjami oprogramowania).	K_W07, Inż_W01																						
Ma wiedzę na temat cyklu życia projektu informatycznego, walidacji i weryfikacji, technik utrzymywania oprogramowania (refaktoryzacji) i potrafi ją stosować w przypadku podstawowych systemów informatycznych.	K_W07, Inż_W01																						
Ma podstawową wiedzę dotyczącą prawnych, społecznych i etycznych aspektów pracy informatyka, ze szczególnym uwzględnieniem zagadnień inżynierii oprogramowania.	K_W07, Inż_W01																						
Ma podstawową wiedzę o strukturze, działaniu i cyklach życia systemów informatycznych oraz o oprogramowaniu urządzeń komputerowych.	K_W07, Inż_W01																						
Potrafi projektować pisać, uruchamiać i testować programy w wybranych paradygmatach i środowiskach programistycznych.	K_U03, Inż_U01																						
Projektuje oprogramowanie zgodnie z metodyką obiektową, posługuje się wzorcami projektowymi i potrafi stworzyć model obiektowy prostego systemu w języku UML.	K_U08, Inż_U05																						
Potrafi wykonać prostą analizę funkcjonowania systemu informatycznego, stworzyć i przeprowadzić plan testowania, wziąć udział w inspekcji oprogramowania.	K_U08, Inż_U04																						
Przy formułowaniu i rozwiązywaniu zadań inżynierskich dostrzega ich aspekty systemowe i pozatechniczne, w szczególności potrafi dokonać wstępnej analizy ekonomicznej podejmowanych działań inżynierskich, w tym szacowania czasochłonności implementowania wybranych metod działania.	K_U07, Inż_U03																						
Jest świadom roli i znaczenia wiedzy w rozwiązywaniu problemów o charakterze poznawczym i praktycznym typowych dla inżynierii oprogramowania.	K_K02																						
Wykazuje gotowość do pełnienia, w sposób odpowiedzialny i respektujący zasady etyki zawodowej, ról zawodowych wymagających kompetencji zdobywanych w ramach studiów na kierunku informatyka, w szczególności jest gotów do formułowania opinii o zagadnieniach związanych z inżynierią oprogramowania.	K_K04																						
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • Jaskiewicz A.: Inżynieria oprogramowania • Sacha K.: Inżynieria oprogramowania • Sommerville i.: Inżynieria oprogramowania • Martin R.C.: Zwinne wytwarzanie oprogramowania. Najlepsze wzorce, zasady i praktyki. • Cohn M.: Agile. Metodyki zwinne w wytwarzaniu oprogramowania. 																						

17.	Metody weryfikacji zakładanych efektów uczenia się egzamin, kolokwium, prezentacja projektu	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu <ul style="list-style-type: none"> • wykład: egzamin (treści weryfikowane na egzaminie, np. modele procesu wytwarzania oprogramowania, zarządzanie jakością, metody testowania), • ćwiczenia: omówienie rozwiązań zadań, • projekt: omówienie dokumentacji projektowej oraz prezentacja fragmentów opracowanej aplikacji 	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	ćwiczenie-pracownia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
	przygotowanie projektu	20 godz.
	przygotowanie do egzaminu	20 godz.
	studiowanie tematyki wykładów i literatury	12 godz.
	przygotowanie do ćwiczeń	30 godz.
	Sumarycznie	
	Łączna liczba godzin	142 godz.
	Liczba punktów ECTS	6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Konstrukcja kompilatorów Compiler construction
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DI1KK
6.	Rodzaj przedmiotu/modułu Informatyczny podstawowy (I1)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenio-pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty • Metody programowania
13.	Cele kształcenia dla przedmiotu Podstawowym celem przedmiotu jest zapoznanie studentów z budową współczesnych kompilatorów oraz z technikami stosowanymi w ich konstruowaniu.
14.	Treści programowe • Wstęp: struktura kompilatora • Leksowanie • Parsowanie • Analiza semantyczna • Generowanie kodu pośredniego • Analiza statyczna i optymalizacja kodu • Generowanie kodu wynikowego • Wsparcie dla języków wysokopoziomowych

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <hr/> <p>po zaliczeniu przedmiotu student będzie znał budowę standardowego kompilatora oraz podstawowe pojęcia i techniki niezbędne do konstruowania poszczególnych modułów, w tym</p> <ul style="list-style-type: none"> • wyrażenia regularne • gramatyki bezkontekstowe • reguły typowania • analizę przepływu danych • techniki optymalizacji kodu <hr/> <p>Umiejętności</p> <hr/> <p>umie skonstruować kompilator prostego języka imperatywnego</p>	<p>K_W05, K_W08</p> <hr/> <p>K_U03, K_U10, K_U13</p>																		
16.	<p>Literatura obowiązkowa i zalecana</p> <p>Modern Compiler Implementation in ML, Appel Compilers: Principles, Techniques, and Tools, Aho Ullman Engineering: A Compiler, Cooper Torczon Advanced Compiler Design and Implementation, Muchnick The Garbage Collection Handbook, Jones</p>																			
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <ul style="list-style-type: none"> • egzamin • projekty programistyczne polegające na doimplementowaniu do przygotowanego szablonu poszczególnych funkcjonalności kompilatora 																			
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Do zaliczenia ćwiczenio-pracownia należy zdobyć wymaganą, podaną w regulaminie przedmiotu liczbę punktów za zadania ćwiczeniowe i pracownie. Punkty w obu wymienionych kategoriach liczą się osobno. Egzamin ma formę pisemną; egzamin poprawkowy może mieć formę ustną. Do jego zaliczenia trzeba uzyskać określoną liczbę punktów.</p>																			
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <hr/> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">ćwiczenio-pracownia</td> <td style="width: 30%;">30 godz.</td> </tr> <tr> <td>wykład</td> <td>30 godz.</td> </tr> </table> <hr/> <p>Praca własna studenta</p> <hr/> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">przygotowanie do ćwiczeń</td> <td style="width: 30%;">10 godz.</td> </tr> <tr> <td>przygotowanie do pracowni</td> <td>10 godz.</td> </tr> <tr> <td>studiowanie tematyki wykładów i literatury</td> <td>10 godz.</td> </tr> <tr> <td>praca nad projektem</td> <td>50 godz.</td> </tr> <tr> <td>przygotowanie do egzaminu</td> <td>10 godz.</td> </tr> </table> <hr/> <p>Sumarycznie</p> <hr/> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Łączna liczba godzin</td> <td style="width: 30%;">150 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>6</td> </tr> </table>		ćwiczenio-pracownia	30 godz.	wykład	30 godz.	przygotowanie do ćwiczeń	10 godz.	przygotowanie do pracowni	10 godz.	studiowanie tematyki wykładów i literatury	10 godz.	praca nad projektem	50 godz.	przygotowanie do egzaminu	10 godz.	Łączna liczba godzin	150 godz.	Liczba punktów ECTS	6
ćwiczenio-pracownia	30 godz.																			
wykład	30 godz.																			
przygotowanie do ćwiczeń	10 godz.																			
przygotowanie do pracowni	10 godz.																			
studiowanie tematyki wykładów i literatury	10 godz.																			
praca nad projektem	50 godz.																			
przygotowanie do egzaminu	10 godz.																			
Łączna liczba godzin	150 godz.																			
Liczba punktów ECTS	6																			

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Metody zarządzania projektami						
2.	Dyscyplina informatyka						
3.	Język wykładowy polski						
4.	Jednostka prowadząca przedmiot Instytut Informatyki						
5.	Kod przedmiotu/modułu 28-INF-S-DI1ZPro						
6.	Rodzaj przedmiotu/modułu Informatyczny podstawowy (I1)						
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka						
8.	Poziom studiów studia I stopnia						
9.	Rok studiów (jeśli obowiązuje) —						
10.	Semestr letni						
11.	Forma zajęć i liczba godzin wykład – 6 godzin						
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak						
13.	Cele przedmiotu Celem wykładów jest wprowadzenie do metodologii zarządzania projektami. Przedstawione zostaną obecne trendy w modelach zarządzania projektami i zasady doboru odpowiedniego modelu w zależności od charakteru projektu.						
14.	Treści programowe <ul style="list-style-type: none"> 1. Model projektu kaskadowego (waterfall) <ul style="list-style-type: none"> • Cykl budowania produktu – od pomysłu do wdrożenia • Etapy realizacji projektu 2. Model projektu zwinnego (agile) <ul style="list-style-type: none"> • Metodyki Scrum, SAFe • Zorganizowanie procesu ciągłych regularnych dostaw (continuous delivery), DevSecOps 3. Zespół projektowy, role i współpraca <ul style="list-style-type: none"> • Projekt z perspektywy dewelopera • Projekt z perspektywy lidera projektu • Projekt z perspektywy klienta 						
15.	Zakładane efekty uczenia się <p>Wiedza</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">zna podstawowe metodologie zarządzania projektami</td> <td style="width: 20%;">K_W08</td> </tr> </table> <p>Umiejętności</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">potrafi rozróżnić podstawowe modele projektów (waterfall & agile)</td> <td style="width: 20%;">K_U10, K_U13</td> </tr> <tr> <td style="width: 80%;">potrafi dobrać właściwą metodologię do prowadzenie projektu</td> <td style="width: 20%;">K_U10, K_U13</td> </tr> </table>	zna podstawowe metodologie zarządzania projektami	K_W08	potrafi rozróżnić podstawowe modele projektów (waterfall & agile)	K_U10, K_U13	potrafi dobrać właściwą metodologię do prowadzenie projektu	K_U10, K_U13
zna podstawowe metodologie zarządzania projektami	K_W08						
potrafi rozróżnić podstawowe modele projektów (waterfall & agile)	K_U10, K_U13						
potrafi dobrać właściwą metodologię do prowadzenie projektu	K_U10, K_U13						
16.	Literatura obowiązkowa i zalecana						
17.	Metody weryfikacji zakładanych efektów kształcenia egzamin						

18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Egzamin testowy	
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela	
	wykład	6 godz.
	Praca własna studenta	
	samodzielne studiowanie materiałów	12 godz.
	przygotowanie do egzaminu lub rozwiązywanie dodatkowych zadań	6 godz.
	Sumarycznie	
	Łączna liczba godzin	24 godz.
	Liczba punktów ECTS	1

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Programowanie funkcyjne Functional Programming
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DP6PF
6.	Rodzaj przedmiotu/modułu Informatyczny podstawowy (I1)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: <ul style="list-style-type: none"> • Metody programowania Niezbędne kompetencje: <ul style="list-style-type: none"> • Umiejętność programowania w języku wyższego poziomu • Pożądana znajomość języka Scheme
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest zapoznanie studentów z mechanizmami i metodami programowania funkcyjnego w językach z typizacją statyczną (OCaml, Haskell). Paradygmat ten obecnie staje się bardzo popularny (również w językach imperatywnych), ponieważ wysoki poziom abstrakcji ułatwia składanie komponentów, a bezstanowość jest wykorzystywana w wysokopoziomowych mechanizmach programowania współbieżnego, które pozwalają pisać efektywne programy, wykorzystujące procesory wielordzeniowe.
14.	Treści programowe <ol style="list-style-type: none"> 1. System typów z polimorfizmem parametrycznym 2. Funkcje wyższych rzędów 3. Algebraiczne typy danych i mechanizm dopasowania do wzorca 4. Ewaluacja gorliwa i leniwa, rekursja ogonowa 5. Efekty obliczeniowe w języku funkcyjnym 6. Abstrakcyjne typy danych, moduły 7. Klasy typów jako mechanizm abstrakcji, monady

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 192 1426 293"> <tr> <td>rozumie specyfikę programowania funkcyjnego</td> <td>K_W05</td> </tr> <tr> <td>zna używane w językach funkcyjnych mechanizmy abstrakcji</td> <td>K_W05</td> </tr> <tr> <td>zna typowe dla programowania funkcyjnego mechanizmy językowe</td> <td>K_W08</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 371 1426 439"> <tr> <td>potrafi wykorzystać mechanizmy paradygmatu funkcyjnego w implementacji programów</td> <td>K_U03</td> </tr> </table>	rozumie specyfikę programowania funkcyjnego	K_W05	zna używane w językach funkcyjnych mechanizmy abstrakcji	K_W05	zna typowe dla programowania funkcyjnego mechanizmy językowe	K_W08	potrafi wykorzystać mechanizmy paradygmatu funkcyjnego w implementacji programów	K_U03						
rozumie specyfikę programowania funkcyjnego	K_W05														
zna używane w językach funkcyjnych mechanizmy abstrakcji	K_W05														
zna typowe dla programowania funkcyjnego mechanizmy językowe	K_W08														
potrafi wykorzystać mechanizmy paradygmatu funkcyjnego w implementacji programów	K_U03														
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • materiały do wykładu, udostępniane przez prowadzącego • Minsky Y., Madhavapeddy A., Hickey J., Real World OCaml, https://v1.realworldocaml.org/ • E.Chailoux, P.Manoury, B.Pagano. Developing Applications with Objective Caml. http://caml.inria.fr/pub/docs/oreilly-book/ • O'Sullivan B., Goerze J., Stewart D., Real World Haskell, http://book.realworldhaskell.org/ 														
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>egzamin pisemny, napisanie programu komputerowego, realizacja zadań przy komputerze</p>														
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Do zaliczenia pracowni konieczne jest zaimplementowanie i wyjaśnienie wymaganych programów, zgodnie z podanymi specyfikacjami (oceny punktowo lub w równoważny sposób).</p> <p>Egzamin ma formę pisemną, aby go zaliczyć konieczne jest zdobycie wymaganej liczby punktów.</p>														
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 1003 1426 1070"> <tr> <td>pracownia</td> <td>30 godz.</td> </tr> <tr> <td>wykład</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 1149 1426 1249"> <tr> <td>przygotowanie do egzaminu</td> <td>20 godz.</td> </tr> <tr> <td>przygotowanie do pracowni</td> <td>40 godz.</td> </tr> <tr> <td>studiowanie tematyki wykładów i literatury</td> <td>30 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 1328 1426 1395"> <tr> <td>Łączna liczba godzin</td> <td>150 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>6</td> </tr> </table>	pracownia	30 godz.	wykład	30 godz.	przygotowanie do egzaminu	20 godz.	przygotowanie do pracowni	40 godz.	studiowanie tematyki wykładów i literatury	30 godz.	Łączna liczba godzin	150 godz.	Liczba punktów ECTS	6
pracownia	30 godz.														
wykład	30 godz.														
przygotowanie do egzaminu	20 godz.														
przygotowanie do pracowni	40 godz.														
studiowanie tematyki wykładów i literatury	30 godz.														
Łączna liczba godzin	150 godz.														
Liczba punktów ECTS	6														

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Programowanie obiektowe Object-oriented programming
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DP6PO
6.	Rodzaj przedmiotu/modułu Informatyczny podstawowy (I1) ze znacznikiem PiPO
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenio-pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Niezbędne kompetencje <ul style="list-style-type: none"> • Podstawowa umiejętność programowania w języku wyższego poziomu
13.	Cele kształcenia dla przedmiotu Głównym celem przedmiotu jest zapoznanie studentów z fundamentami programowania obiektowego: klasa, obiekt, dziedziczenie, metody wirtualne, ukrywanie implementacji czy wzorce projektowe, oraz praktyczne ich wykorzystanie w programowaniu. W trakcie zajęć będą też omówione metody projektowania oprogramowania wykorzystujące techniki obiektowe.
14.	Treści programowe <ul style="list-style-type: none"> • Różne style programowania • Klasy i obiekty, konstruktory, dziedziczenie • Metody wirtualne, przeciążanie metod • Programowanie generyczne • Dziedziczenie a implementacja interfejsu • Wybrane wzorce projektowe: singleton, template, strategy • Modelowanie obiektowe za pomocą notacji UML • Analiza i projektowanie obiektowe • Paradygmaty programowania wywodzące się z programowania obiektowego • Obiekty rozproszone i trwałość obiektów • Dobre praktyki programowania obiektowego: zasada podstawienia Liskov, zasada pojedynczej odpowiedzialności

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <hr/> <p>zna i rozumie podstawowe mechanizmy obiektowe (dziedziczenie, enkapsulacja, abstrakcja, metody wirtualne) K_W05, Inż_W01</p> <p>zna podstawy UML w zakresie diagramów klas K_W05, Inż_W01</p> <p>zna przynajmniej 2 języki obiektowe w stopniu podstawowym K_W05, Inż_W01</p> <p>zna podstawy projektowania obiektowego K_W05, Inż_W01</p> <hr/> <p>Umiejętności</p> <hr/> <p>umie korzystać z obiektowych klas bibliotecznych i tworzyć własne klasy biblioteczne K_U03, Inż_U04</p> <p>potrafi określić i korzystać z podstawowych zależności między obiektami (wzorce projektowe) K_U07, Inż_U04</p> <p>umie zaprojektować obiektowo aplikację K_U08, Inż_U01</p> <p>umie opracować i przedstawić swój projekt K_U08, Inż_U05</p>																
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • M. Weisfeld; Myślenie obiektowe w programowaniu • H. Ledgard; Mała księga programowania obiektowego • Peter Coad, Edward Yourdon; Object-oriented analysis • B. Meyer; Programowanie zorientowane obiektowo • I. Graham; Metody obiektowe w teorii i praktyce 																
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>zaprogramowanie i prezentacja programu komputerowego, implementacja i prezentacja większego projektu</p>																
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Do zaliczenia ćwiczenio-pracowni należy zdobyć wskazaną w regulaminie liczbę punktów za implementację zadań.</p> <p>Do zaliczenia egzaminu konieczna jest realizacja projektu.</p>																
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <hr/> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">ćwiczenio-pracownia</td> <td style="width: 30%;">30 godz.</td> </tr> <tr> <td>wykład</td> <td>30 godz.</td> </tr> </table> <hr/> <p>Praca własna studenta</p> <hr/> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">samodzielna implementacja zadań</td> <td style="width: 30%;">35 godz.</td> </tr> <tr> <td>prezentacja projektu</td> <td>1 godz.</td> </tr> <tr> <td>studiowanie dokumentacji i literatury dodatkowej</td> <td>20 godz.</td> </tr> <tr> <td>opracowanie i implementacja projektu</td> <td>30 godz.</td> </tr> </table> <hr/> <p>Sumarycznie</p> <hr/> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Łączna liczba godzin</td> <td style="width: 30%;">146 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>6</td> </tr> </table>	ćwiczenio-pracownia	30 godz.	wykład	30 godz.	samodzielna implementacja zadań	35 godz.	prezentacja projektu	1 godz.	studiowanie dokumentacji i literatury dodatkowej	20 godz.	opracowanie i implementacja projektu	30 godz.	Łączna liczba godzin	146 godz.	Liczba punktów ECTS	6
ćwiczenio-pracownia	30 godz.																
wykład	30 godz.																
samodzielna implementacja zadań	35 godz.																
prezentacja projektu	1 godz.																
studiowanie dokumentacji i literatury dodatkowej	20 godz.																
opracowanie i implementacja projektu	30 godz.																
Łączna liczba godzin	146 godz.																
Liczba punktów ECTS	6																

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Programowanie współbieżne Concurrent programming
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-PrgWspl
6.	Rodzaj przedmiotu/modułu Informatyczny podstawowy (I1)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład – 30 godzin, ćwiczenie-pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: <ul style="list-style-type: none"> • zalecany Wstęp do informatyki Niezbędne kompetencje: <ul style="list-style-type: none"> • podstawy programowania w języku Java, np. zdobyte podczas Kursu tego języka.
13.	Cele przedmiotu Zapoznanie studentów z podstawami programowania współbieżnego i wnioskowania o poprawności programów współbieżnych.
14.	Treści programowe <ol style="list-style-type: none"> 1. Problem sekcji krytycznej i jego pierwsze rozwiązania 2. Współbieżne obiekty 3. Pamięć dzielona 4. Blokady wirujące i przetrzymywanie 5. Monitory 6. Listy, kolejki i stosy współbieżne 7. Bariery 8. Model pamięci języka Java 9. Programowanie transakcyjne

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 192 1161 510"> <tr> <td>zna i rozumie problemy, jakie powstają przy współbieżnym dostępie do współdzielonych danych</td> <td>K_W05</td> </tr> <tr> <td>zna metody specyfikacji algorytmów/obiektów współbieżnych, a w szczególności pojęcie linearyzacji</td> <td>K_W05</td> </tr> <tr> <td>zna środki synchronizacji, zarówno niskopoziomowe (instrukcje maszynowe) jak i wysokopoziomowe (oferowane przez system operacyjny)</td> <td>K_W07</td> </tr> <tr> <td>zna znaczenie wsparcia ze strony maszyny do zaimplementowania środków synchronizacji we współczesnych architekturach sprzętowych</td> <td>K_W07</td> </tr> <tr> <td>zna praktyczne, efektywne obliczeniowo współbieżne implementacje podstawowych struktur danych: liczników, list, kolejek, stosów, tablic haszujących</td> <td>K_W08</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 595 1161 846"> <tr> <td>potrafi zweryfikować poprawność prostych algorytmów współbieżnych, uzasadnić spełnienie warunków bezpieczeństwa (wskazać punkty linearyzacji) i żywotności (brak głodzenia, niewstrzymywanie, niehamowanie)</td> <td>K_U10, K_U13</td> </tr> <tr> <td>potrafi ocenić wydajność implementacji algorytmu/obiektu współbieżnego, odwołując się do wiedzy o współczesnych architekturach sprzętowych</td> <td>K_U03, K_U10, K_U13</td> </tr> <tr> <td>potrafi dobrać środek synchronizacji do zastosowania oraz zaprogramować niewstrzymujące lub niehamujące implementacje podstawowych struktur danych i ich wariantów</td> <td>K_U10, K_U13</td> </tr> </table>	zna i rozumie problemy, jakie powstają przy współbieżnym dostępie do współdzielonych danych	K_W05	zna metody specyfikacji algorytmów/obiektów współbieżnych, a w szczególności pojęcie linearyzacji	K_W05	zna środki synchronizacji, zarówno niskopoziomowe (instrukcje maszynowe) jak i wysokopoziomowe (oferowane przez system operacyjny)	K_W07	zna znaczenie wsparcia ze strony maszyny do zaimplementowania środków synchronizacji we współczesnych architekturach sprzętowych	K_W07	zna praktyczne, efektywne obliczeniowo współbieżne implementacje podstawowych struktur danych: liczników, list, kolejek, stosów, tablic haszujących	K_W08	potrafi zweryfikować poprawność prostych algorytmów współbieżnych, uzasadnić spełnienie warunków bezpieczeństwa (wskazać punkty linearyzacji) i żywotności (brak głodzenia, niewstrzymywanie, niehamowanie)	K_U10, K_U13	potrafi ocenić wydajność implementacji algorytmu/obiektu współbieżnego, odwołując się do wiedzy o współczesnych architekturach sprzętowych	K_U03, K_U10, K_U13	potrafi dobrać środek synchronizacji do zastosowania oraz zaprogramować niewstrzymujące lub niehamujące implementacje podstawowych struktur danych i ich wariantów	K_U10, K_U13
zna i rozumie problemy, jakie powstają przy współbieżnym dostępie do współdzielonych danych	K_W05																
zna metody specyfikacji algorytmów/obiektów współbieżnych, a w szczególności pojęcie linearyzacji	K_W05																
zna środki synchronizacji, zarówno niskopoziomowe (instrukcje maszynowe) jak i wysokopoziomowe (oferowane przez system operacyjny)	K_W07																
zna znaczenie wsparcia ze strony maszyny do zaimplementowania środków synchronizacji we współczesnych architekturach sprzętowych	K_W07																
zna praktyczne, efektywne obliczeniowo współbieżne implementacje podstawowych struktur danych: liczników, list, kolejek, stosów, tablic haszujących	K_W08																
potrafi zweryfikować poprawność prostych algorytmów współbieżnych, uzasadnić spełnienie warunków bezpieczeństwa (wskazać punkty linearyzacji) i żywotności (brak głodzenia, niewstrzymywanie, niehamowanie)	K_U10, K_U13																
potrafi ocenić wydajność implementacji algorytmu/obiektu współbieżnego, odwołując się do wiedzy o współczesnych architekturach sprzętowych	K_U03, K_U10, K_U13																
potrafi dobrać środek synchronizacji do zastosowania oraz zaprogramować niewstrzymujące lub niehamujące implementacje podstawowych struktur danych i ich wariantów	K_U10, K_U13																
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • Maurice Herlihy, Nir Shavit, Victor Luchangco, Michael Spear: The Art of Multiprocessor Programming, 2nd ed. • Brian Goetz, Tim Peierls, Joshua Bloch et al.: Java Concurrency in Practice 																
17.	<p>Metody weryfikacji zakładanych efektów kształcenia</p> <p>Prezentacja rozwiązania zadania, napisanie i prezentacja programu komputerowego, egzamin pisemny, referat.</p>																
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Warunkiem zaliczenia ćwiczeń jest uzyskanie minimum punktowego za przygotowane rozwiązania zadań na podstawie deklaracji weryfikowanych w trakcie ćwiczeń. Aby zaliczyć wykład, należy zdobyć co najmniej połowę punktów z egzaminu końcowego. Szczegółowe warunki zaliczenia przedmiotu są określone w regulaminie przedmiotu.</p>																
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 1364 1426 1435"> <tr> <td>wykład</td> <td>30 godz.</td> </tr> <tr> <td>ćwiczenie-pracownia</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 1507 1426 1641"> <tr> <td>przygotowanie do egzaminu</td> <td>15 godz.</td> </tr> <tr> <td>samodzielne rozwiązywanie zadań ćwiczeniowych i programistycznych</td> <td>50 godz.</td> </tr> <tr> <td>udział w egzaminie</td> <td>2 godz.</td> </tr> <tr> <td>przygotowywanie się do ćwiczeń (w tym lektura literatury)</td> <td>25 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 1720 1426 1785"> <tr> <td>Łączna liczba godzin</td> <td>152 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>6</td> </tr> </table>	wykład	30 godz.	ćwiczenie-pracownia	30 godz.	przygotowanie do egzaminu	15 godz.	samodzielne rozwiązywanie zadań ćwiczeniowych i programistycznych	50 godz.	udział w egzaminie	2 godz.	przygotowywanie się do ćwiczeń (w tym lektura literatury)	25 godz.	Łączna liczba godzin	152 godz.	Liczba punktów ECTS	6
wykład	30 godz.																
ćwiczenie-pracownia	30 godz.																
przygotowanie do egzaminu	15 godz.																
samodzielne rozwiązywanie zadań ćwiczeniowych i programistycznych	50 godz.																
udział w egzaminie	2 godz.																
przygotowywanie się do ćwiczeń (w tym lektura literatury)	25 godz.																
Łączna liczba godzin	152 godz.																
Liczba punktów ECTS	6																

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Rachunek prawdopodobieństwa i statystyka Probability and Statistics
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DP6RPS
6.	Rodzaj przedmiotu/modułu Informatyczny podstawowy (I1) ze znacznikiem RPiS
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty <ul style="list-style-type: none"> • Algebra • Analiza matematyczna Niezbędne kompetencje: <ul style="list-style-type: none"> • Wstępna znajomość rachunku całkowego • Znajomość operacji macierzowych
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest: <ul style="list-style-type: none"> • Powiązanie intuicyjnego pojęcia prawdopodobieństwa z formalnymi określeniami. • Zapoznanie słuchaczy z typowymi rozkładami probabilistycznymi. • Część zasadnicza - przejście od próbki pochodzącej ze znanego rozkładu do statystyk decyzyjnych. • Podstawowe informacje o analizie wariancji i regresji liniowej, związku między tymi sposobami analizowania danych.

14.	<p>Treści programowe</p> <ol style="list-style-type: none"> 1. Przestrzeń probabilistyczna, pojęcie zmiennej losowej, jej rozkład i opis. 2. Przykłady zmiennych losowych typu dyskretnego i typu ciągłego. Charakterystyki zmiennych losowych - momenty. 3. Rozkłady zmiennych losowych wielowymiarowych (rozkład dwuwymiarowy, rozkład warunkowy, rozkład brzegowy, niezależność dwóch zmiennych losowych); Macierze kowariancji i korelacji. Wielowymiarowy rozkład normalny i szczególny przypadek dwuwymiarowy. 4. Funkcje dwuwymiarowych zmiennych losowych. Wyznaczanie gęstości i dystrybuanty funkcji zmiennych losowych. 5. Funkcja charakterystyczna i jej własności. Związek funkcji charakterystycznej z momentami zmiennej. 6. Populacja i próbka. Rozkłady którym podlega próbka (chi-kwadrat, t-Studenta, F). Centralne twierdzenie graniczne. 7. Wielowymiarowy rozkład normalny. 8. Regresja liniowa i analiza wariancji - wstęp. 																														
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 689 1426 1041"> <tr> <td>rozumie pojęcia zmiennej losowej i przestrzeni probabilistycznej</td> <td>K_W01</td> </tr> <tr> <td>zna podstawowe charakterystyki opisowe zmiennej losowej</td> <td>K_W01</td> </tr> <tr> <td>rozumie pojęcie niezależności zmiennych losowych i potrafi połączyć to z rozkładem łącznym (wielowymiarowym)</td> <td>K_W01</td> </tr> <tr> <td>zna pojęcie funkcji tworzącej momenty</td> <td>K_W01</td> </tr> <tr> <td>rozumie zasady testowania hipotez, zna pojęcia statystyki testowej, obszaru krytycznego</td> <td>K_W01</td> </tr> <tr> <td>posiada przeglądową wiedzę o różnych rodzajach regresji oraz o analizie wariancji</td> <td>K_W01</td> </tr> <tr> <td>rozumie znaczenie obliczeń statystycznych w wymiarze ekonomicznym i kulturowym</td> <td>K_W01</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 1124 1426 1384"> <tr> <td>potrafi obliczyć wielkości opisowe zmiennej losowej: wartość oczekiwaną, wariancję, korelację</td> <td>K_U01</td> </tr> <tr> <td>umie wyznaczyć rozkład łączny i funkcje zmiennych losowych</td> <td>K_U01</td> </tr> <tr> <td>potrafi wykorzystać funkcję tworzącą momenty do identyfikacji rozkładu</td> <td>K_U01</td> </tr> <tr> <td>zna typowe testy parametryczne i rozumie ich konstrukcję</td> <td>K_U01</td> </tr> <tr> <td>posiada wiedzę o związku regresji z aproksymacją średniokwadratową</td> <td>K_U01</td> </tr> <tr> <td>umie przeprowadzić typowe rodzaje analizy wariancji</td> <td>K_U01</td> </tr> <tr> <td>rozumie związek analizy wariancji z regresją liniową</td> <td>K_U01</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 1467 1426 1503"> <tr> <td>jest świadomy zagrożeń wynikających z gromadzenia danych</td> <td>K_K02, K_K04</td> </tr> </table>	rozumie pojęcia zmiennej losowej i przestrzeni probabilistycznej	K_W01	zna podstawowe charakterystyki opisowe zmiennej losowej	K_W01	rozumie pojęcie niezależności zmiennych losowych i potrafi połączyć to z rozkładem łącznym (wielowymiarowym)	K_W01	zna pojęcie funkcji tworzącej momenty	K_W01	rozumie zasady testowania hipotez, zna pojęcia statystyki testowej, obszaru krytycznego	K_W01	posiada przeglądową wiedzę o różnych rodzajach regresji oraz o analizie wariancji	K_W01	rozumie znaczenie obliczeń statystycznych w wymiarze ekonomicznym i kulturowym	K_W01	potrafi obliczyć wielkości opisowe zmiennej losowej: wartość oczekiwaną, wariancję, korelację	K_U01	umie wyznaczyć rozkład łączny i funkcje zmiennych losowych	K_U01	potrafi wykorzystać funkcję tworzącą momenty do identyfikacji rozkładu	K_U01	zna typowe testy parametryczne i rozumie ich konstrukcję	K_U01	posiada wiedzę o związku regresji z aproksymacją średniokwadratową	K_U01	umie przeprowadzić typowe rodzaje analizy wariancji	K_U01	rozumie związek analizy wariancji z regresją liniową	K_U01	jest świadomy zagrożeń wynikających z gromadzenia danych	K_K02, K_K04
rozumie pojęcia zmiennej losowej i przestrzeni probabilistycznej	K_W01																														
zna podstawowe charakterystyki opisowe zmiennej losowej	K_W01																														
rozumie pojęcie niezależności zmiennych losowych i potrafi połączyć to z rozkładem łącznym (wielowymiarowym)	K_W01																														
zna pojęcie funkcji tworzącej momenty	K_W01																														
rozumie zasady testowania hipotez, zna pojęcia statystyki testowej, obszaru krytycznego	K_W01																														
posiada przeglądową wiedzę o różnych rodzajach regresji oraz o analizie wariancji	K_W01																														
rozumie znaczenie obliczeń statystycznych w wymiarze ekonomicznym i kulturowym	K_W01																														
potrafi obliczyć wielkości opisowe zmiennej losowej: wartość oczekiwaną, wariancję, korelację	K_U01																														
umie wyznaczyć rozkład łączny i funkcje zmiennych losowych	K_U01																														
potrafi wykorzystać funkcję tworzącą momenty do identyfikacji rozkładu	K_U01																														
zna typowe testy parametryczne i rozumie ich konstrukcję	K_U01																														
posiada wiedzę o związku regresji z aproksymacją średniokwadratową	K_U01																														
umie przeprowadzić typowe rodzaje analizy wariancji	K_U01																														
rozumie związek analizy wariancji z regresją liniową	K_U01																														
jest świadomy zagrożeń wynikających z gromadzenia danych	K_K02, K_K04																														
16.	<p>Literatura obowiązkowa i zalecana</p> <ol style="list-style-type: none"> 1. Tadeusz Gernstenkorn, Tadeusz Śródka: Kombinatoryka i rachunek prawdopodobieństwa. PWN, 1974. 2. Małgorzata Majsnerowska: Wprowadzenie do rachunku prawdopodobieństwa z zadaniami. Wydawnictwo BTC, 2009. 3. Mieczysław Sobczyk: Statystyka. Wydawnictwo Uniwersytetu Marii Curie-Skłodowskiej, 1994. 4. Jacek Jakubowski, Rafał Sztencel: Wstęp do teorii prawdopodobieństwa. Script, 2001. 																														
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <ul style="list-style-type: none"> • Listy zadań - około 10 zadań tygodniowo. • Zadania do obliczeń komputerowych. • Kolokwium. • Egzamin. 																														
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Zaliczenie wykładu - egzamin, zadania obliczeniowe w trakcie semestru.</p> <p>Zaliczenie ćwiczeń - listy zadań, kolokwium w trakcie semestru.</p>																														

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	ćwiczenia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
	przygotowanie do ćwiczeń	45 godz.
	praca nad zadaniami obliczeniowymi	5 godz.
	przygotowanie do egzaminu	15 godz.
	kolokwium	3 godz.
	przygotowanie do kolokwium	10 godz.
	czytanie dodatkowych materiałów	10 godz.
	egzamin	3 godz.
Sumarycznie		
Łączna liczba godzin	151 godz.	
Liczba punktów ECTS	6	

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Sieci komputerowe Computer networks
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DP6SK
6.	Rodzaj przedmiotu/modułu Informatyczny podstawowy (I1) ze znacznikiem SK
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykładu – 30 godzin, ćwiczenio-pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: <ul style="list-style-type: none"> • Algebra lub Matematyka dyskretna • Systemy operacyjne lub Systemy komputerowe Niezbędne kompetencje: <ul style="list-style-type: none"> • Znajomość grup i pierścieni. • Znajomość tematyki procesów, sygnałów i komunikacji międzyprocesowej. • Obsługa wiersza poleceń w systemie Linux, umiejętność pisania prostych skryptów w powłoce. • Dobra umiejętność programowania w C lub C++.
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest zaznajomienie słuchaczy z mechanizmami, protokołami i algorytmicznymi podstawami efektywnej i bezpiecznej komunikacji w sieciach komputerowych, ze szczególnym uwzględnieniem Internetu.

14.	<p>Treści programowe</p> <ol style="list-style-type: none"> 1. Podstawowe pojęcia, protokoły, model warstwowy. 2. Warstwa sieciowa: adresowanie CIDR, routing, protokoły IP i ICMP, tworzenie tablic routingu oparte o stan łączy i wektory odległości, systemy autonomiczne, budowa routera, NAT. 3. Warstwa łącza danych i fizyczna: Ethernet, sieci bezprzewodowe, dostęp do współdzielonego nośnika (CSMA/CD, CSMA/CA). 4. Warstwa transportowa: niezawodny transport, mechanizmy okna przesuwne, kontrola przepływu, kontrola przeciążenia, protokoły UDP i TCP. 5. Warstwa zastosowań: DNS, HTTP, poczta elektroniczna, sieci P2P, serwery proxy, współpraca z NAT. 6. Programowanie gniazd: gniazda surowe, datagramowe i strumieniowe. 7. Podstawy kodowania i kryptografii: kody detekcyjne (CRC) i korekcyjne, szyfrowanie, podpisy cyfrowe, RSA, SSL, PGP. 8. Podstawy bezpieczeństwa sieci. 																
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 658 1423 913"> <tr> <td data-bbox="225 658 1166 779">Zna podstawowe mechanizmy, protokoły i algorytmiczne podstawy sieci komputerowych: model warstwowy, routing, model klient-serwer, niezawodny transport, mechanizmy okna przesuwne, kontrola przepływu, kontrola przeciążenia, protokoły dostępu do współdzielonego medium.</td> <td data-bbox="1174 658 1423 779">K_W07, Inż_W01</td> </tr> <tr> <td data-bbox="225 786 1166 846">Zna implementacje powyższych mechanizmów we współczesnym Internecie: protokoły TCP, UDP, IP, Ethernet.</td> <td data-bbox="1174 786 1423 846">K_W07, Inż_W01</td> </tr> <tr> <td data-bbox="225 853 1166 913">Zna elementy kryptografii, podstawowe słabości stosowanych protokołów i typy ataków sieciowych.</td> <td data-bbox="1174 853 1423 913">K_W07, Inż_W01</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 999 1423 1223"> <tr> <td data-bbox="225 999 1166 1059">Konfiguruje działanie prostej sieci lokalnej złożonej z komputerów, przełączników i routerów.</td> <td data-bbox="1174 999 1423 1059">K_U09, Inż_U02</td> </tr> <tr> <td data-bbox="225 1066 1166 1126">Jest w stanie zdiagnozować i rozwiązać problemy z brakiem połączenia, połączeniem złej jakości czy też wadliwą konfiguracją routingu.</td> <td data-bbox="1174 1066 1423 1126">K_U09, Inż_U01</td> </tr> <tr> <td data-bbox="225 1133 1166 1193">Posługuje się standardowymi narzędziami szyfrowania danych, potrafi podsłuchiwać nieszyfrowany ruch sieciowy.</td> <td data-bbox="1174 1133 1423 1193">K_U09, Inż_U04</td> </tr> <tr> <td data-bbox="225 1200 1166 1223">Pisze programy komunikujące się przez sieć wykorzystując interfejs gniazd.</td> <td data-bbox="1174 1200 1423 1223">K_U03, Inż_U05</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 1308 1423 1368"> <tr> <td data-bbox="225 1308 1166 1368">Rozumie znaczenie Internetu w funkcjonowaniu współczesnego społeczeństwa, rozumie możliwości i zagrożenia z tym związane.</td> <td data-bbox="1174 1308 1423 1368">K_K02, K_K04</td> </tr> </table>	Zna podstawowe mechanizmy, protokoły i algorytmiczne podstawy sieci komputerowych: model warstwowy, routing, model klient-serwer, niezawodny transport, mechanizmy okna przesuwne, kontrola przepływu, kontrola przeciążenia, protokoły dostępu do współdzielonego medium.	K_W07, Inż_W01	Zna implementacje powyższych mechanizmów we współczesnym Internecie: protokoły TCP, UDP, IP, Ethernet.	K_W07, Inż_W01	Zna elementy kryptografii, podstawowe słabości stosowanych protokołów i typy ataków sieciowych.	K_W07, Inż_W01	Konfiguruje działanie prostej sieci lokalnej złożonej z komputerów, przełączników i routerów.	K_U09, Inż_U02	Jest w stanie zdiagnozować i rozwiązać problemy z brakiem połączenia, połączeniem złej jakości czy też wadliwą konfiguracją routingu.	K_U09, Inż_U01	Posługuje się standardowymi narzędziami szyfrowania danych, potrafi podsłuchiwać nieszyfrowany ruch sieciowy.	K_U09, Inż_U04	Pisze programy komunikujące się przez sieć wykorzystując interfejs gniazd.	K_U03, Inż_U05	Rozumie znaczenie Internetu w funkcjonowaniu współczesnego społeczeństwa, rozumie możliwości i zagrożenia z tym związane.	K_K02, K_K04
Zna podstawowe mechanizmy, protokoły i algorytmiczne podstawy sieci komputerowych: model warstwowy, routing, model klient-serwer, niezawodny transport, mechanizmy okna przesuwne, kontrola przepływu, kontrola przeciążenia, protokoły dostępu do współdzielonego medium.	K_W07, Inż_W01																
Zna implementacje powyższych mechanizmów we współczesnym Internecie: protokoły TCP, UDP, IP, Ethernet.	K_W07, Inż_W01																
Zna elementy kryptografii, podstawowe słabości stosowanych protokołów i typy ataków sieciowych.	K_W07, Inż_W01																
Konfiguruje działanie prostej sieci lokalnej złożonej z komputerów, przełączników i routerów.	K_U09, Inż_U02																
Jest w stanie zdiagnozować i rozwiązać problemy z brakiem połączenia, połączeniem złej jakości czy też wadliwą konfiguracją routingu.	K_U09, Inż_U01																
Posługuje się standardowymi narzędziami szyfrowania danych, potrafi podsłuchiwać nieszyfrowany ruch sieciowy.	K_U09, Inż_U04																
Pisze programy komunikujące się przez sieć wykorzystując interfejs gniazd.	K_U03, Inż_U05																
Rozumie znaczenie Internetu w funkcjonowaniu współczesnego społeczeństwa, rozumie możliwości i zagrożenia z tym związane.	K_K02, K_K04																
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • James F. Kurose, Keith W. Ross, Sieci komputerowe. Ujęcie całościowe, Helion, 2018. • Andrew S. Tanenbaum, Sieci komputerowe, Helion, 2012. • W. Richard Stevens, Unix programowanie usług sieciowych, tom 1, WNT, 2002, rozdziały 1-9 i 25. 																
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>prezentacja rozwiązania zadania, napisanie programu komputerowego, realizacja zadań przy komputerze, egzamin pisemny</p>																
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Warunkiem zaliczenia ćwiczenio-pracowni jest uzyskanie minimum 40 punktów łącznie za pisemne przygotowanie zadań domowych, aktywną realizację zadań warsztatowych przy komputerze i za projekty programistyczne. Co najmniej 10 punktów należy uzyskać za realizację projektów programistycznych. Warunkiem zaliczenia wykładu jest zdobycie co najmniej 60 punktów na egzaminie pisemnym. Szczegółowe warunki zaliczenia przedmiotu określone są w regulaminie przedmiotu.</p>																

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	ćwiczenie-pracownia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
	rozwiązywanie zadań domowych	15 godz.
	studiowanie tematyki wykładu oraz przygotowanie do egzaminu	20 godz.
	pisanie projektów programistycznych	40 godz.
	udział w egzaminie	3 godz.
	przygotowywanie się do zajęć warsztatowych	15 godz.
Sumarycznie		
Łączna liczba godzin	153 godz.	
Liczba punktów ECTS	6	

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Systemy komputerowe Computer Systems
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DP14ASKSO
6.	Rodzaj przedmiotu/modułu Informatyczny podstawowy (I1) ze znacznikami ASK oraz SO
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: <ul style="list-style-type: none"> • Logika dla informatyków • Wstęp do informatyki lub Algorytmy i struktury danych Pomocna będzie elementarna znajomość języka C.
13.	Cele kształcenia dla przedmiotu Celem wykładu jest przedstawienie studentom najważniejszych algorytmów i struktur danych wykorzystywanych w systemach komputerowych, głównie w sprzęcie, systemach operacyjnych i narzędziach systemowych.

14.	<p>Treści programowe</p> <ol style="list-style-type: none"> 1. Sprzętowa implementacja operacji arytmetycznych: szybkie algorytmy dodawania (sumatory carry-lookahead i Ling'a), mnożenia (drzewa Wallace'a i kodowanie Booth'a) i dzielenia (non-restoring division, SRT division). 2. Niskopoziomowa reprezentacja programów (kod trójadresowy, graf przepływu sterowania), analiza przepływu danych (obliczenia stałopunktowe). 3. Architektura współczesnego procesora (przetwarzanie potokowe i out-of-order). Graf przepływu danych i algorytm Tomasulo. 4. Spekulatywne wykonywanie instrukcji, przewidywanie skoków (perceptron, TAGE), atak Spectre. 5. Instrumentacja kodu, profilowanie (graf wywołań funkcji) i optymalizacja kodu (algorytm local value numbering). 6. Struktury pamięci podręcznych, algorytmy cache-oblivious. 7. Algorytmy szeregowania wątków i balansowania obciążenia procesorów (algorytm ULE, kolejka kalendarzowa). 8. Translacja adresów, wielopoziomowa i odwrócona tablica stron. 9. Pamięć wirtualna, algorytmy zastępowania stron (algorytm WSCLOCK, aging, Clock-Pro). 10. Algorytmy dynamicznego przydziału pamięci (buddy systems, boundary tags). Algorytmy odświeżania (mark and sweep, algorytm kopiujący). 11. Struktury danych systemów plików (grupy bloków, i-node, B-drzewo, journaling). Planowanie operacji I/O (budget-fair queueing). 12. Przetwarzanie współbieżne. Implementacja środków synchronizacji (synchronizacja pamięci podręcznych, exponential backoff). 13. Kontrola przepływności w protokole TCP. Obliczanie sum kontrolnych i kodów korekcyjnych. 14. Sieci w układach scalonych: topologia i trasowanie pakietów. 																				
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 976 1426 1171"> <tr> <td data-bbox="225 976 1166 1070">Wie jak programy komunikują się ze sprzętem i systemem operacyjnym</td> <td data-bbox="1174 976 1426 1005">K_W07, Inż_W01</td> </tr> <tr> <td data-bbox="225 1070 1166 1171">Rozumie wpływ architektury maszyny na implementację kompilatorów i systemów operacyjnych</td> <td data-bbox="1174 1070 1426 1099">K_W07, Inż_W01</td> </tr> <tr> <td data-bbox="225 1171 1166 1200">Posiada przeglądową wiedzę na temat komponentów systemu operacyjnego.</td> <td data-bbox="1174 1171 1426 1200">K_W07, Inż_W01</td> </tr> <tr> <td data-bbox="225 1200 1166 1294">Zna mechanizmy, polityki i algorytmy, które wykorzystują jądra systemów operacyjnych w celu realizacji usług dla programów użytkownika.</td> <td data-bbox="1174 1200 1426 1229">K_W07, Inż_W01</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 1252 1426 1574"> <tr> <td data-bbox="225 1252 1166 1317">Umie pisać programy zachowujące się poprawnie w obliczu ograniczeń maszynowej reprezentacji typów numerycznych</td> <td data-bbox="1174 1252 1426 1281">K_U03, Inż_U05</td> </tr> <tr> <td data-bbox="225 1317 1166 1382">Potrafi wybrać i zastosować właściwe narzędzia i metodologie do mierzenia wydajności kodu programów</td> <td data-bbox="1174 1317 1426 1382">K_U09, Inż_U1, Inż_U04</td> </tr> <tr> <td data-bbox="225 1382 1166 1447">Potrafi przeprowadzić analizę kodu pod względem wykorzystania pamięci podręcznej i przeorganizować go celem optymalizacji</td> <td data-bbox="1174 1382 1426 1447">K_U09, Inż_U2, Inż_U04</td> </tr> <tr> <td data-bbox="225 1447 1166 1512">Potrafi użyć środków synchronizacji do konstrukcji poprawnych programów współbieżnych.</td> <td data-bbox="1174 1447 1426 1476">K_U09, Inż_U05</td> </tr> <tr> <td data-bbox="225 1512 1166 1574">Pomaga członkom zespołu w pisaniu oprogramowania wolnego od usterek związanych z niskopoziomowymi aspektami działania komputerów</td> <td data-bbox="1174 1512 1426 1541">K_U14</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 1655 1426 1720"> <tr> <td data-bbox="225 1655 1166 1720">Potrafi właściwie wskazać te części projektu komercyjnego, które oplaca się optymalizować</td> <td data-bbox="1174 1655 1426 1684">K_K02, K_K04</td> </tr> </table>	Wie jak programy komunikują się ze sprzętem i systemem operacyjnym	K_W07, Inż_W01	Rozumie wpływ architektury maszyny na implementację kompilatorów i systemów operacyjnych	K_W07, Inż_W01	Posiada przeglądową wiedzę na temat komponentów systemu operacyjnego.	K_W07, Inż_W01	Zna mechanizmy, polityki i algorytmy, które wykorzystują jądra systemów operacyjnych w celu realizacji usług dla programów użytkownika.	K_W07, Inż_W01	Umie pisać programy zachowujące się poprawnie w obliczu ograniczeń maszynowej reprezentacji typów numerycznych	K_U03, Inż_U05	Potrafi wybrać i zastosować właściwe narzędzia i metodologie do mierzenia wydajności kodu programów	K_U09, Inż_U1, Inż_U04	Potrafi przeprowadzić analizę kodu pod względem wykorzystania pamięci podręcznej i przeorganizować go celem optymalizacji	K_U09, Inż_U2, Inż_U04	Potrafi użyć środków synchronizacji do konstrukcji poprawnych programów współbieżnych.	K_U09, Inż_U05	Pomaga członkom zespołu w pisaniu oprogramowania wolnego od usterek związanych z niskopoziomowymi aspektami działania komputerów	K_U14	Potrafi właściwie wskazać te części projektu komercyjnego, które oplaca się optymalizować	K_K02, K_K04
Wie jak programy komunikują się ze sprzętem i systemem operacyjnym	K_W07, Inż_W01																				
Rozumie wpływ architektury maszyny na implementację kompilatorów i systemów operacyjnych	K_W07, Inż_W01																				
Posiada przeglądową wiedzę na temat komponentów systemu operacyjnego.	K_W07, Inż_W01																				
Zna mechanizmy, polityki i algorytmy, które wykorzystują jądra systemów operacyjnych w celu realizacji usług dla programów użytkownika.	K_W07, Inż_W01																				
Umie pisać programy zachowujące się poprawnie w obliczu ograniczeń maszynowej reprezentacji typów numerycznych	K_U03, Inż_U05																				
Potrafi wybrać i zastosować właściwe narzędzia i metodologie do mierzenia wydajności kodu programów	K_U09, Inż_U1, Inż_U04																				
Potrafi przeprowadzić analizę kodu pod względem wykorzystania pamięci podręcznej i przeorganizować go celem optymalizacji	K_U09, Inż_U2, Inż_U04																				
Potrafi użyć środków synchronizacji do konstrukcji poprawnych programów współbieżnych.	K_U09, Inż_U05																				
Pomaga członkom zespołu w pisaniu oprogramowania wolnego od usterek związanych z niskopoziomowymi aspektami działania komputerów	K_U14																				
Potrafi właściwie wskazać te części projektu komercyjnego, które oplaca się optymalizować	K_K02, K_K04																				
16.	<p>Literatura obowiązkowa i zalecana</p> <p>Wybrane przez wykładowcę fragmenty poniższych publikacji:</p> <ul style="list-style-type: none"> • Hennessy, John L., and David A. Patterson. Computer Architecture: A Quantitative Approach (6th ed.). • Israel Koren, Computer Arithmetic Algorithms (2nd ed.). • Flemming Nielson, Hanne R. Nielson, and Chris Hankin. Principles of Program Analysis. • D. Harris, S. Harris, Digital Design and Computer Architecture (2nd ed.) • Randal Bryant, David O'Hallaron. Computer Systems: A Programmer's Perspective (3rd ed.). • Bruce Jacob, Spencer Ng, and David Wang. Memory Systems: Cache, DRAM, Disk (2nd ed.) • Abraham Silberschatz, Peter Baer Galvin, Greg Gagne. Operating System Concepts (10th ed.) • Maurice J. Bach. The design of the UNIX operating system. 																				

17.	Metody weryfikacji zakładanych efektów uczenia się prezentacja rozwiązań zadań, egzamin pisemny																
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Warunkiem zaliczenia ćwiczeń jest uzyskanie minimum punktowego za przygotowane rozwiązania zadań (na podstawie deklaracji weryfikowanych w trakcie ćwiczeń). Aby zaliczyć wykład, należy zdobyć co najmniej połowę punktów z końcowego egzaminu pisemnego. Szczegółowe warunki zaliczenia przedmiotu są określone w regulaminie przedmiotu.																
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">ćwiczenia</td> <td style="width: 20%;">30 godz.</td> </tr> <tr> <td>wykład</td> <td>30 godz.</td> </tr> </table> Praca własna studenta <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">przygotowanie do egzaminu</td> <td style="width: 20%;">25 godz.</td> </tr> <tr> <td>egzamin</td> <td>2 godz.</td> </tr> <tr> <td>rozwiązywania zadań z list</td> <td>30 godz.</td> </tr> <tr> <td>analiza i weryfikacja notatek z zajęć oraz innych źródeł informacji</td> <td>30 godz.</td> </tr> </table> Sumarycznie <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Łączna liczba godzin</td> <td style="width: 20%;">147 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>6</td> </tr> </table>	ćwiczenia	30 godz.	wykład	30 godz.	przygotowanie do egzaminu	25 godz.	egzamin	2 godz.	rozwiązywania zadań z list	30 godz.	analiza i weryfikacja notatek z zajęć oraz innych źródeł informacji	30 godz.	Łączna liczba godzin	147 godz.	Liczba punktów ECTS	6
ćwiczenia	30 godz.																
wykład	30 godz.																
przygotowanie do egzaminu	25 godz.																
egzamin	2 godz.																
rozwiązywania zadań z list	30 godz.																
analiza i weryfikacja notatek z zajęć oraz innych źródeł informacji	30 godz.																
Łączna liczba godzin	147 godz.																
Liczba punktów ECTS	6																

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Systemy operacyjne Operating Systems
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DI1SO
6.	Rodzaj przedmiotu/modułu Informatyczny podstawowy (I1) ze znacznikiem SO
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin, pracownia — 15 godzin, repetytorium — 15 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty <ul style="list-style-type: none"> • Kurs: Podstawowy warsztat informatyka • Kurs: Wstęp do programowania w języku C • Architektury systemów komputerowych Niezbędne kompetencje <ul style="list-style-type: none"> • Umiejętność obsługi wiersza poleceń systemu uniksowego • Dobra znajomość języka C • Podstawowa wiedza o przerwaniach i wyjątkach procesora, tłumaczeniu adresów i pamięci wirtualnej oraz konsolidacji
13.	Cele kształcenia dla przedmiotu Celem wykładu jest zapoznanie słuchaczy z wybranymi aspektami systemu uniksowego z punktu widzenia programisty systemowego i administratora. Na przykładzie wywołań systemowych, biblioteki języka C i narzędzi standardu POSIX.1 zostaną przedstawione najważniejsze osiągnięcia w dziedzinie systemów operacyjnych.

14.	<p>Treści programowe</p> <ol style="list-style-type: none"> 1. Jądro, wywołania systemowe, zmiana kontekstu, wyłączenie. 2. Procesy: stan i środowisko wykonania, tworzenie i kończenie, oczekiwanie na zakończenie, ładowanie programów i hierarchia. 3. Sygnały i nielokalne skoki. 4. Zarządzanie zadaniami, sesje, terminal, organizacja powłoki uniksowej. 5. Pliki: rodzaje, operacje, ścieżki, pliki urządzeń, deskryptory plików. 6. System plików: struktury dyskowe. 7. Buforowanie plików. 8. Tożsamość, autoryzacja i uwierzytelnianie. 9. Komunikacja międzyprocesowa. 10. Pamięć wirtualna, odwzorowania plików w pamięć. 11. Zarządzanie stertą: algorytmy przydziału pamięci. 12. Programowanie sieciowe TCP/IP, architektura klient-serwer. 13. Wielowątkowość i problemy z współbieżnością. 14. Synchronizacja: muteksy i zmienne warunkowe. 																						
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 723 1426 853"> <tr> <td data-bbox="225 723 1166 763">Posiada przeglądową wiedzę na temat komponentów systemu operacyjnego.</td> <td data-bbox="1171 723 1426 763">K_W07, Inż_W01</td> </tr> <tr> <td data-bbox="225 763 1166 813">Zna mechanizmy, polityki i algorytmy, które wykorzystują jądra SO w celu realizacji usług dla programów użytkownika.</td> <td data-bbox="1171 763 1426 813">K_W07, Inż_W01</td> </tr> <tr> <td data-bbox="225 813 1166 853">Rozumie wpływ architektury systemu komputerowego na implementację SO.</td> <td data-bbox="1171 813 1426 853">K_W07, Inż_W01</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 936 1426 1317"> <tr> <td data-bbox="225 936 1166 999">Potrafi użyć środków synchronizacji do konstrukcji poprawnych programów współbieżnych.</td> <td data-bbox="1171 936 1426 999">K_U03, Inż_U05</td> </tr> <tr> <td data-bbox="225 999 1166 1061">Umie wykorzystać abstrakcje oferowane przez jądro SO przy projektowaniu oprogramowania.</td> <td data-bbox="1171 999 1426 1061">K_U03, Inż_U04</td> </tr> <tr> <td data-bbox="225 1061 1166 1124">Potrafi czytać i analizować kod jądra SO z użyciem wyspecjalizowanych narzędzi.</td> <td data-bbox="1171 1061 1426 1124">K_U09, Inż_U04</td> </tr> <tr> <td data-bbox="225 1124 1166 1187">Umie śledzić interakcję programów użytkownika z jądrem SO w celu znajdowania usterek i problemów z wydajnością.</td> <td data-bbox="1171 1124 1426 1187">K_U09, Inż_U01</td> </tr> <tr> <td data-bbox="225 1187 1166 1249">Potrafi zaproponować bezpieczną i wydajną metodę przechowywania danych z użyciem usług SO.</td> <td data-bbox="1171 1187 1426 1249">K_U09</td> </tr> <tr> <td data-bbox="225 1249 1166 1317">W projektowaniu oprogramowania posługuje się zasadą separacji mechanizmu od polityki.</td> <td data-bbox="1171 1249 1426 1317">K_U09</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 1400 1426 1529"> <tr> <td data-bbox="225 1400 1166 1462">Potrafi ocenić przydatność danego SO do implementacji projektu w pracy zawodowej.</td> <td data-bbox="1171 1400 1426 1462">K_K02, K_K04</td> </tr> <tr> <td data-bbox="225 1462 1166 1529">Pomaga członkom zespołu wybrać właściwe interfejsy i usługi SO do realizacji zadania.</td> <td data-bbox="1171 1462 1426 1529">K_K02</td> </tr> </table>	Posiada przeglądową wiedzę na temat komponentów systemu operacyjnego.	K_W07, Inż_W01	Zna mechanizmy, polityki i algorytmy, które wykorzystują jądra SO w celu realizacji usług dla programów użytkownika.	K_W07, Inż_W01	Rozumie wpływ architektury systemu komputerowego na implementację SO.	K_W07, Inż_W01	Potrafi użyć środków synchronizacji do konstrukcji poprawnych programów współbieżnych.	K_U03, Inż_U05	Umie wykorzystać abstrakcje oferowane przez jądro SO przy projektowaniu oprogramowania.	K_U03, Inż_U04	Potrafi czytać i analizować kod jądra SO z użyciem wyspecjalizowanych narzędzi.	K_U09, Inż_U04	Umie śledzić interakcję programów użytkownika z jądrem SO w celu znajdowania usterek i problemów z wydajnością.	K_U09, Inż_U01	Potrafi zaproponować bezpieczną i wydajną metodę przechowywania danych z użyciem usług SO.	K_U09	W projektowaniu oprogramowania posługuje się zasadą separacji mechanizmu od polityki.	K_U09	Potrafi ocenić przydatność danego SO do implementacji projektu w pracy zawodowej.	K_K02, K_K04	Pomaga członkom zespołu wybrać właściwe interfejsy i usługi SO do realizacji zadania.	K_K02
Posiada przeglądową wiedzę na temat komponentów systemu operacyjnego.	K_W07, Inż_W01																						
Zna mechanizmy, polityki i algorytmy, które wykorzystują jądra SO w celu realizacji usług dla programów użytkownika.	K_W07, Inż_W01																						
Rozumie wpływ architektury systemu komputerowego na implementację SO.	K_W07, Inż_W01																						
Potrafi użyć środków synchronizacji do konstrukcji poprawnych programów współbieżnych.	K_U03, Inż_U05																						
Umie wykorzystać abstrakcje oferowane przez jądro SO przy projektowaniu oprogramowania.	K_U03, Inż_U04																						
Potrafi czytać i analizować kod jądra SO z użyciem wyspecjalizowanych narzędzi.	K_U09, Inż_U04																						
Umie śledzić interakcję programów użytkownika z jądrem SO w celu znajdowania usterek i problemów z wydajnością.	K_U09, Inż_U01																						
Potrafi zaproponować bezpieczną i wydajną metodę przechowywania danych z użyciem usług SO.	K_U09																						
W projektowaniu oprogramowania posługuje się zasadą separacji mechanizmu od polityki.	K_U09																						
Potrafi ocenić przydatność danego SO do implementacji projektu w pracy zawodowej.	K_K02, K_K04																						
Pomaga członkom zespołu wybrać właściwe interfejsy i usługi SO do realizacji zadania.	K_K02																						
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • Operating Systems: Three Easy Pieces Remzi Andrea Arpaci-Dusseau • Advanced Programming in the UNIX Environment (3rd Edition) Richard Stevens, Stephen Argo 																						
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>egzamin pisemny, projekt programistyczny, prezentacja rozwiązania zadania</p>																						
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Do zaliczenia ćwiczeń i pracowni należy zdobyć co najmniej połowę liczby punktów za zadania ćwiczeniowe i programistyczne. Na pracowni będą co najmniej trzy projekty programistyczne do wykonania: powłoka uniksowa, algorytm przydziału pamięci, system plików. Na punkty z poszczególnych aktywności mogą być narzucone minimalne progi zaliczeniowe.</p> <p>Egzamin końcowy jest w formie pisemnej i wymaga zdobycia co najmniej połowy punktów. Zadania mogą mieć następujące formy: opisanie problemu, prezentacja działania algorytmu, test wielokrotnego wyboru.</p>																						

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	repetitorium	15 godz.
	pracownia	15 godz.
	ćwiczenia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
	udział w egzaminie	3 godz.
	przygotowanie do egzaminu	10 godz.
	rozwiązywanie zadań z list	60 godz.
rozwiązywanie zadań programistycznych	50 godz.	
Sumarycznie		
Łączna liczba godzin	213 godz.	
Liczba punktów ECTS	8	

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Testowanie gier Game Testing
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-TestGier
6.	Rodzaj przedmiotu/modułu Informatyczny podstawowy (I1)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenio-pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zaliczone przedmioty: zaliczony kurs jednego z języków programowania Niezbędne kompetencje: elementarna znajomość języka C++, Java. Większość zadań programistycznych wymaga podstawowej znajomości Javy.
13.	Cele kształcenia dla przedmiotu Celem przedmiotu Testowanie gier jest pomoc w doskonaleniu umiejętności wystarczająco dobrego testowania gier komputerowych.
14.	Treści programowe Wykłady <ul style="list-style-type: none"> • Podstawy testowania gier • Poziomy i typy testów • Dwie podstawowe zasady testowania gier • Rola testera gier • Znaczenie wystarczająco dobrego testowania gry • Jakość gry komputerowej • Plan testowania gry komputerowej • Proces testowania gry • Testing by the Numbers, Combinational Testing • Test Flow Diagrams • Cleanroom Testing, Test Tree • Ad Hoc Testing oraz Gameplay testing • Defect Triggers • Regression testing and Test Reuse • Exploratory Game Testing

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="228 192 1157 450"> <tr> <td>ma poszerzoną, uporządkowaną i podbudowaną teoretycznie wiedzę w obrębie testowania gier</td> <td>K_W08</td> </tr> <tr> <td>zna i rozumie pojęcia: specyfikacji gry, poprawność i testowalności wymagań, oraz zna zasady planowania wykonania testów gry</td> <td>K_W08</td> </tr> <tr> <td>ma podstawową wiedzę na temat tworzenia i weryfikacji oraz walidacji przypadków testowych</td> <td>K_W08</td> </tr> <tr> <td>zna podstawowe metody projektowania i analizy podstawy testów</td> <td>K_W05</td> </tr> <tr> <td>zna podstawowe metody planowania procesu testowego gry</td> <td>K_W08</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="228 533 1157 790"> <tr> <td>potrafi testować gry</td> <td>K_U03, K_U10</td> </tr> <tr> <td>umie określić cele testowania gry</td> <td>K_U10</td> </tr> <tr> <td>potrafi definiować i analizować testowalność wymagań gry określonych przez jej specyfikacje</td> <td>K_U10, K_U13</td> </tr> <tr> <td>analizuje poprawność i złożoność przypadku testowego i procesu testowego</td> <td>K_U10</td> </tr> <tr> <td>umie stosować podstawowe metody i typy testów gry</td> <td>K_U10</td> </tr> <tr> <td>ma umiejętność używania popularnych automatycznych narzędzi testowania gier</td> <td>K_U10</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="228 873 1157 936"> <tr> <td>jest świadom roli i znaczenia wiedzy na temat testowania oprogramowania, w szczególności gier komputerowych</td> <td>K_K06</td> </tr> </table>	ma poszerzoną, uporządkowaną i podbudowaną teoretycznie wiedzę w obrębie testowania gier	K_W08	zna i rozumie pojęcia: specyfikacji gry, poprawność i testowalności wymagań, oraz zna zasady planowania wykonania testów gry	K_W08	ma podstawową wiedzę na temat tworzenia i weryfikacji oraz walidacji przypadków testowych	K_W08	zna podstawowe metody projektowania i analizy podstawy testów	K_W05	zna podstawowe metody planowania procesu testowego gry	K_W08	potrafi testować gry	K_U03, K_U10	umie określić cele testowania gry	K_U10	potrafi definiować i analizować testowalność wymagań gry określonych przez jej specyfikacje	K_U10, K_U13	analizuje poprawność i złożoność przypadku testowego i procesu testowego	K_U10	umie stosować podstawowe metody i typy testów gry	K_U10	ma umiejętność używania popularnych automatycznych narzędzi testowania gier	K_U10	jest świadom roli i znaczenia wiedzy na temat testowania oprogramowania, w szczególności gier komputerowych	K_K06	
ma poszerzoną, uporządkowaną i podbudowaną teoretycznie wiedzę w obrębie testowania gier	K_W08																									
zna i rozumie pojęcia: specyfikacji gry, poprawność i testowalności wymagań, oraz zna zasady planowania wykonania testów gry	K_W08																									
ma podstawową wiedzę na temat tworzenia i weryfikacji oraz walidacji przypadków testowych	K_W08																									
zna podstawowe metody projektowania i analizy podstawy testów	K_W05																									
zna podstawowe metody planowania procesu testowego gry	K_W08																									
potrafi testować gry	K_U03, K_U10																									
umie określić cele testowania gry	K_U10																									
potrafi definiować i analizować testowalność wymagań gry określonych przez jej specyfikacje	K_U10, K_U13																									
analizuje poprawność i złożoność przypadku testowego i procesu testowego	K_U10																									
umie stosować podstawowe metody i typy testów gry	K_U10																									
ma umiejętność używania popularnych automatycznych narzędzi testowania gier	K_U10																									
jest świadom roli i znaczenia wiedzy na temat testowania oprogramowania, w szczególności gier komputerowych	K_K06																									
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • strony www w języku angielskim ISTQB (International Software Testing Qualifications Board) Certified Tester Foundation Level (CTFL) i wyższe poziomy • strony www Stowarzyszenia Jakości Systemów Informatycznych • książka: Schultz Ch. P., Bryant R. D.: Game Testing All in One, Third Edition Mercury Learning and information, 2017 																									
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <ul style="list-style-type: none"> • prezentacja rozwiązań zadań podczas ćwiczeń i w pracowni, • weryfikacja i walidacja zadań przygotowanych i wykonanych przez studentów, • test pisemny i egzamin. 																									
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Aby uzyskać zaliczenie ćwiczeń i pracowni, student musi zebrać określoną minimalną liczbę punktów do rozwiązania zestawów zadań (na podstawie deklaracji zweryfikowanych podczas ćwiczeń/pracowni) oraz uzyskać pozytywne oceny z dwóch kolokwii. Aby zaliczyć przedmiot, należy zaliczyć ćwiczenia i pracownię i uzyskać co najmniej 60 punktów za końcowy egzamin pisemny.</p>																									
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="228 1585 1157 1659"> <tr> <td>ćwiczenie-pracownia</td> <td>30 godz.</td> </tr> <tr> <td>wykład</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="228 1731 1157 1895"> <tr> <td>testy i egzamin</td> <td>6 godz.</td> </tr> <tr> <td>rozwiązywanie zestawów zadań</td> <td>30 godz.</td> </tr> <tr> <td>przygotowanie się do testu</td> <td>4 godz.</td> </tr> <tr> <td>przygotowanie się do egzaminu</td> <td>16 godz.</td> </tr> <tr> <td>lektura materiału</td> <td>25 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="228 1977 1157 2045"> <tr> <td>Łączna liczba godzin</td> <td>141 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>6</td> </tr> </table>	ćwiczenie-pracownia	30 godz.	wykład	30 godz.	testy i egzamin	6 godz.	rozwiązywanie zestawów zadań	30 godz.	przygotowanie się do testu	4 godz.	przygotowanie się do egzaminu	16 godz.	lektura materiału	25 godz.	Łączna liczba godzin	141 godz.	Liczba punktów ECTS	6							
ćwiczenie-pracownia	30 godz.																									
wykład	30 godz.																									
testy i egzamin	6 godz.																									
rozwiązywanie zestawów zadań	30 godz.																									
przygotowanie się do testu	4 godz.																									
przygotowanie się do egzaminu	16 godz.																									
lektura materiału	25 godz.																									
Łączna liczba godzin	141 godz.																									
Liczba punktów ECTS	6																									

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Wstęp do bezpieczeństwa komputerowego Introduction to Computer Security
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DP6WBK
6.	Rodzaj przedmiotu/modułu Informatyczny podstawowy (I1)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 30 godzin, ćwiczenie-pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Niezbędne kompetencje: <ul style="list-style-type: none">• podstawowa znajomość protokołów sieciowych
13.	Cele przedmiotu Podstawowym celem przedmiotu jest nauka tworzenia bezpiecznych aplikacji, w szczególności doboru odpowiednich schematów kryptograficznych. Studenci poznają zagadnienia związane z analizą kodu pod kątem znajdowania podatności.
14.	Treści programowe Program przedmiotu: <ul style="list-style-type: none">• Podstawowe pojęcia: zagrożenia, polityki bezpieczeństwa, mechanizmy bezpieczeństwa.• Podstawowe schematy kryptograficzne, modelowanie atakującego.• Bezpieczeństwo aplikacji webowych m. in. ataki na: uwierzytelnianie, zarządzanie sesją, autoryzację, SQL-injection.• Ataki niskopoziomowe (m. in. buffer overflows).• Bezpieczeństwo przechowywania haseł (memory-hard functions), uwierzytelnianie dwuskładnikowe.• Ataki typu side-channel (m. in. timing-attacks, cache attacks, padding-oracle attacks).• Prywatność i anonimowość (differential privacy; losowość a bezpieczeństwo; fingerprinting; bezpieczeństwo aplikacji mobilnych).• Wykorzystanie metod statystycznych do wykrywania incydentów bezpieczeństwa

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 192 1426 259"> <tr> <td>zna metody zapewniające poufność komunikacji</td> <td>K_W08</td> </tr> <tr> <td>zna metody zapewniające integralność komunikacji</td> <td>K_W08</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 342 1426 439"> <tr> <td>umie modelować zagrożenia w postaci definiowania polityk bezpieczeństwa</td> <td>K_U10</td> </tr> <tr> <td>umie stosować właściwe algorytmy kryptograficzne w celu zabezpieczenia systemu komputerowego przed zagrożeniami</td> <td>K_U10, K_U13</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 521 1426 555"> <tr> <td>potrafi dobrać adekwatne narzędzia w celu tworzenia bezpiecznych aplikacji</td> <td>K_K06</td> </tr> </table>	zna metody zapewniające poufność komunikacji	K_W08	zna metody zapewniające integralność komunikacji	K_W08	umie modelować zagrożenia w postaci definiowania polityk bezpieczeństwa	K_U10	umie stosować właściwe algorytmy kryptograficzne w celu zabezpieczenia systemu komputerowego przed zagrożeniami	K_U10, K_U13	potrafi dobrać adekwatne narzędzia w celu tworzenia bezpiecznych aplikacji	K_K06				
zna metody zapewniające poufność komunikacji	K_W08														
zna metody zapewniające integralność komunikacji	K_W08														
umie modelować zagrożenia w postaci definiowania polityk bezpieczeństwa	K_U10														
umie stosować właściwe algorytmy kryptograficzne w celu zabezpieczenia systemu komputerowego przed zagrożeniami	K_U10, K_U13														
potrafi dobrać adekwatne narzędzia w celu tworzenia bezpiecznych aplikacji	K_K06														
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • M. Sajdak i in., Bezpieczeństwo aplikacji webowych, 2022 • R. Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems, 2020 • M. Bishop, Computer Security. Art and Science, 2018 • Mark Dowd, John McDonald, Justin Schuh, The Art of Software Security Assessment – Identifying and Preventing Software Vulnerabilities • Gynvael Coldwind, Zrozumieć programowanie • Douglas R. Stinson, Cryptography. Theory and practice • Mike Rosulek, The Joy of Cryptography 														
17.	<p>Metody weryfikacji zakładanych efektów kształcenia</p> <ul style="list-style-type: none"> • egzamin pisemny, • prezentacja rozwiązanych zadań, • napisanie programów komputerowych realizujących wybrane ataki 														
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Do zaliczenia ćwiczenio-pracowni należy zdobyć wymaganą, podaną w regulaminie przedmiotu liczbę punktów za zadania ćwiczeniowe, pracowniowe i opcjonalny projekt. Punkty za wszystkie wyżej wymienione aktywności liczą się łącznie.</p> <p>Egzamin ma formę pisemną, aby go zaliczyć konieczne jest zdobycie wymaganej liczby punktów. Osoby, które osiągnęły bardzo dobre wyniki na ćwiczeniach i zdobyły ustaloną liczbę punktów za rozwiązanie dodatkowych, trudniejszych zadań mogą uzyskać zwolnienie z egzaminu.</p>														
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 1346 1426 1442"> <tr> <td>wykład</td> <td>30 godz.</td> </tr> <tr> <td>ćwiczenio-pracownia</td> <td>30 godz.</td> </tr> <tr> <td>udział w egzaminie</td> <td>2 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 1525 1426 1592"> <tr> <td>samodzielne rozwiązywanie zadań pracownianych</td> <td>45 godz.</td> </tr> <tr> <td>przygotowywanie się do ćwiczeń (w tym czytanie materiałów dodatkowych)</td> <td>45 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 1675 1426 1733"> <tr> <td>Łączna liczba godzin</td> <td>152 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>6</td> </tr> </table>	wykład	30 godz.	ćwiczenio-pracownia	30 godz.	udział w egzaminie	2 godz.	samodzielne rozwiązywanie zadań pracownianych	45 godz.	przygotowywanie się do ćwiczeń (w tym czytanie materiałów dodatkowych)	45 godz.	Łączna liczba godzin	152 godz.	Liczba punktów ECTS	6
wykład	30 godz.														
ćwiczenio-pracownia	30 godz.														
udział w egzaminie	2 godz.														
samodzielne rozwiązywanie zadań pracownianych	45 godz.														
przygotowywanie się do ćwiczeń (w tym czytanie materiałów dodatkowych)	45 godz.														
Łączna liczba godzin	152 godz.														
Liczba punktów ECTS	6														

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Wstęp do informatyki Introduction to computer science
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DP6WI
6.	Rodzaj przedmiotu/modułu Informatyczny podstawowy (I1)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Elementarna znajomość języka C lub Python (począwszy od wykładu 5)
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest przedstawienie struktury logicznej i zasad działania komputera, cyfrowej reprezentacji informacji oraz sposobu wykonywania programów przez komputer. Istotnym celem jest też umożliwienie studentom nabycia podstawowych umiejętności z zakresu konstruowania i analizy algorytmów, w tym weryfikacji poprawności i szacowania złożoności obliczeniowej. W ramach przedmiotu umożliwia się poznanie podstawowych struktur danych i ich implementacji, a także nabycie umiejętności stosowania takich struktur danych do rozwiązywanych problemów.

14.	<p>Treści programowe</p> <ul style="list-style-type: none"> • Pojęcia: problem, specyfikacja, algorytm, program komputerowy. Budowa komputera – schemat logiczny. • Elementy programowania w języku niskopoziomowym na przykładzie maszyny RAM • Reprezentacja danych w pamięci komputera: binarna stałopozycyjna i zmiennopozycyjna reprezentacja liczb, arytmetyka binarna; reprezentacja tekstu, obrazu, dźwięku. • Programowanie w wysokopoziomym języku strukturalnym, zstępująca metoda programowania. • Kompilacja, translacja, interpretacja programów. Zarządzanie pamięcią podczas uruchamiania programu. • Podstawowe konstrukcje programistyczne: iteracja, zagnieżdżanie, licznik, sumator, rekurencja. • Elementy analizy algorytmów: poprawność, własność stopu, złożoność czasowa i pamięciowa, miary złożoności; notacja asymptotyczna. • Podstawowe struktury danych i ich zastosowania: tablica, lista wiązana, stos, kolejka; drzewo przeszukiwań binarnych. Abstrakcyjne typy danych. • Sortowanie przy pomocy porównań. • Podstawowe techniki algorytmiczne: metoda nawrotów, metoda dziel i zwyciężaj, programowanie dynamiczne, algorytmy zachłanne. • Komputerowa reprezentacja grafów. • Algorytmy grafowe: przeszukiwanie w głąb, wszerz; składowe spójności. • Gramatyki bezkontekstowe jako narzędzie do opisu składni języków programowania. Notacja BNF i EBNF. • Pojęcie rozstrzygalności problemów, przykłady problemów nierozstrzygalnych. 																				
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="220 913 1426 1137"> <tr> <td>zna i rozumie pojęcie specyfikacji problemu algorytmicznego, zgodności algorytmu ze specyfikacją i sposobu wykonywania programów przez komputer</td> <td>K_W06</td> </tr> <tr> <td>posiada podstawową wiedzę na temat cyfrowej reprezentacji danych - liczby, tekst, obrazy, grafy</td> <td>K_W08</td> </tr> <tr> <td>zna podstawowe metody projektowania i analizy algorytmów</td> <td>K_W08</td> </tr> <tr> <td>zna podstawowe struktury danych i wykonywane na nich operacje, a także koncepcję abstrakcyjnych struktur danych</td> <td>K_W05</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="220 1220 1426 1541"> <tr> <td>potrafi formułować precyzyjną specyfikację problemu</td> <td>K_U06</td> </tr> <tr> <td>konstruuje algorytmy rozwiązujące standardowe problemy sformułowane w postaci specyfikacji</td> <td>K_U06</td> </tr> <tr> <td>potrafi analizować poprawność i złożoność obliczeniową prostych algorytmów, w tym rekurencyjnych</td> <td>K_U06</td> </tr> <tr> <td>wykorzystuje podstawowe techniki algorytmiczne, struktury danych i obiekty abstrakcyjne (np. grafy) do rozwiązywania problemów</td> <td>K_U06</td> </tr> <tr> <td>potrafi interpretować i zapisywać elementy składni języków programowania w postaci gramatyk bezkontekstowych</td> <td>K_U03</td> </tr> <tr> <td>posługuje się cyfrową reprezentacją danych: liczb, obrazów, grafów</td> <td>K_U13</td> </tr> </table>	zna i rozumie pojęcie specyfikacji problemu algorytmicznego, zgodności algorytmu ze specyfikacją i sposobu wykonywania programów przez komputer	K_W06	posiada podstawową wiedzę na temat cyfrowej reprezentacji danych - liczby, tekst, obrazy, grafy	K_W08	zna podstawowe metody projektowania i analizy algorytmów	K_W08	zna podstawowe struktury danych i wykonywane na nich operacje, a także koncepcję abstrakcyjnych struktur danych	K_W05	potrafi formułować precyzyjną specyfikację problemu	K_U06	konstruuje algorytmy rozwiązujące standardowe problemy sformułowane w postaci specyfikacji	K_U06	potrafi analizować poprawność i złożoność obliczeniową prostych algorytmów, w tym rekurencyjnych	K_U06	wykorzystuje podstawowe techniki algorytmiczne, struktury danych i obiekty abstrakcyjne (np. grafy) do rozwiązywania problemów	K_U06	potrafi interpretować i zapisywać elementy składni języków programowania w postaci gramatyk bezkontekstowych	K_U03	posługuje się cyfrową reprezentacją danych: liczb, obrazów, grafów	K_U13
zna i rozumie pojęcie specyfikacji problemu algorytmicznego, zgodności algorytmu ze specyfikacją i sposobu wykonywania programów przez komputer	K_W06																				
posiada podstawową wiedzę na temat cyfrowej reprezentacji danych - liczby, tekst, obrazy, grafy	K_W08																				
zna podstawowe metody projektowania i analizy algorytmów	K_W08																				
zna podstawowe struktury danych i wykonywane na nich operacje, a także koncepcję abstrakcyjnych struktur danych	K_W05																				
potrafi formułować precyzyjną specyfikację problemu	K_U06																				
konstruuje algorytmy rozwiązujące standardowe problemy sformułowane w postaci specyfikacji	K_U06																				
potrafi analizować poprawność i złożoność obliczeniową prostych algorytmów, w tym rekurencyjnych	K_U06																				
wykorzystuje podstawowe techniki algorytmiczne, struktury danych i obiekty abstrakcyjne (np. grafy) do rozwiązywania problemów	K_U06																				
potrafi interpretować i zapisywać elementy składni języków programowania w postaci gramatyk bezkontekstowych	K_U03																				
posługuje się cyfrową reprezentacją danych: liczb, obrazów, grafów	K_U13																				
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman; Algorytmy i struktury danych; Helion, [rozdziały 1-3, 5.1 – 5.2, 6, 8.1-8.3, 9, 10] • David Harel, Rzecz o istocie informatyki – algorytmika, WNT, 2008. [rozdziały 1 – 6, częściowo 7 i 8] • A.V. Aho, J.D. Ullman, Wykłady z informatyki z przykładami w języku C, Helion, Gliwice 2003 [rozdziały 1-3, 5, 6, 9.1 – 9.4, 11.1 – 11.4] • Ryuhei Uehara, First Course in Algorithms Through Puzzles, Springer, 2019. [rozdziały 1 - 5, poza 4.4] 																				
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>prezentacja rozwiązania zadania, kolokwium, egzamin pisemny</p>																				

18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Warunkiem zaliczenia ćwiczeń jest uzyskanie minimum punktowego łącznie za przygotowane rozwiązania zadań (na podstawie deklaracji weryfikowanych w trakcie ćwiczeń) oraz dwa kolokwia. Aby zaliczyć wykład, należy zdobyć co najmniej połowę punktów z końcowego egzaminu pisemnego. Szczegółowe warunki zaliczenia przedmiotu są określone w regulaminie przedmiotu.</p>																				
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 376 1428 443"> <tr> <td data-bbox="225 376 1166 409">ćwiczenia</td> <td data-bbox="1166 376 1428 409">30 godz.</td> </tr> <tr> <td data-bbox="225 409 1166 443">wykład</td> <td data-bbox="1166 409 1428 443">30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 521 1428 712"> <tr> <td data-bbox="225 521 1166 555">udział w kolokwiach i egzaminie</td> <td data-bbox="1166 521 1428 555">5 godz.</td> </tr> <tr> <td data-bbox="225 555 1166 589">analiza i weryfikacja notatek z zajęć oraz innych źródeł informacji</td> <td data-bbox="1166 555 1428 589">15 godz.</td> </tr> <tr> <td data-bbox="225 589 1166 622">nabycie umiejętności posługiwania się symulatorem maszyny RAM –</td> <td data-bbox="1166 589 1428 622">5 godz.</td> </tr> <tr> <td data-bbox="225 622 1166 656">przygotowanie do kolokwiów</td> <td data-bbox="1166 622 1428 656">15 godz.</td> </tr> <tr> <td data-bbox="225 656 1166 689">przygotowanie do egzaminu</td> <td data-bbox="1166 656 1428 689">20 godz.</td> </tr> <tr> <td data-bbox="225 689 1166 712">rozwiązywanie zadań z list</td> <td data-bbox="1166 689 1428 712">30 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 790 1428 857"> <tr> <td data-bbox="225 790 1166 824">Łączna liczba godzin</td> <td data-bbox="1166 790 1428 824">150 godz.</td> </tr> <tr> <td data-bbox="225 824 1166 857">Liczba punktów ECTS</td> <td data-bbox="1166 824 1428 857">6</td> </tr> </table>	ćwiczenia	30 godz.	wykład	30 godz.	udział w kolokwiach i egzaminie	5 godz.	analiza i weryfikacja notatek z zajęć oraz innych źródeł informacji	15 godz.	nabycie umiejętności posługiwania się symulatorem maszyny RAM –	5 godz.	przygotowanie do kolokwiów	15 godz.	przygotowanie do egzaminu	20 godz.	rozwiązywanie zadań z list	30 godz.	Łączna liczba godzin	150 godz.	Liczba punktów ECTS	6
ćwiczenia	30 godz.																				
wykład	30 godz.																				
udział w kolokwiach i egzaminie	5 godz.																				
analiza i weryfikacja notatek z zajęć oraz innych źródeł informacji	15 godz.																				
nabycie umiejętności posługiwania się symulatorem maszyny RAM –	5 godz.																				
przygotowanie do kolokwiów	15 godz.																				
przygotowanie do egzaminu	20 godz.																				
rozwiązywanie zadań z list	30 godz.																				
Łączna liczba godzin	150 godz.																				
Liczba punktów ECTS	6																				

3 Przedmioty informatyczne inżynierskie (Inż)

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Komunikacja człowiek-komputer Human-Computer Interaction
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DP9HCI
6.	Rodzaj przedmiotu/modułu Informatyczny inżynierski (Inż)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenio-pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty <ul style="list-style-type: none"> • Przedmiot „Komunikacja człowiek-komputer” jest wybitnie interdyscyplinarny (zob. „Opis przedmiotu” powyżej sylabusu). Niezbędne lub pożądanane kompetencje <ul style="list-style-type: none"> • Zainteresowana osoba powinna zdradzać otwartość na humanistyczne aspekty najwyższej warstwy oprogramowania i zastosowań informatyki. • Zainteresowana osoba powinna umieć i (lub) chcieć analizować i poddawać refleksji przedmioty w otoczeniu, w tym zewnętrzną otoczkę programów komputerowych. • Zainteresowana osoba powinna być dociekliwa i umieć dociekać „prawd” o użyteczności przedmiotów (w tym programów komputerowych) w kontaktach z potencjalnymi, rzeczywistymi, typowymi lub wyrafinowanymi użytkownikami i użytkownikami. • Prócz oczywistych kwalifikacji informatycznych, w tym technik inżynierii oprogramowania (osobny wykład) i niektórych aspektów AI (odrębny wykład), zainteresowania psychologią, socjologią, antropologią, lingwistyką, fizjologią, ergonomią, wzornictwem przemysłowym, a nawet filozofią i sztuką dają osobie studiującej KCK (HCI) przewagę twórczą i produkcyjną w czasie zajęć. • Pożądana jest elementarna znajomość któregoś z języków tworzenia interaktywnych stron WWW (np. HTML, osobny wykład), aplikacji okienkowych (inny wykład), a także wprawne operowanie którymś z systemów konstruowania makiet interfejsów (ćwiczeniom i opanowywanym samodzielnie jako „wartość dodana” 22 zajęć). • Elementarna wiedza o cyklu wytwarzania oprogramowania (osobny wykład z inżynierii oprogramowania).

13.	<p>Cele kształcenia dla przedmiotu</p> <p>Uczestniczka lub uczestnik kursu ma okazję poznać elementy składające się na dobry interfejs, w rozumieniu wywierania jak najkorzystniejszych wrażeń na użytkownika docelowym, opanować podstawową wiedzę psychologiczną i socjologiczną w zakresie przydatnym w projektowaniu GUI, zrozumieć związki między udanym interfejsem a sukcesem ekonomicznym, poznać podstawowe zasady ergonomii i komunikacji werbalnej i wizualnej (a także dźwiękowej) z zastosowaniem w interfejsach komputerowych, reguły operowania przestrzenią, barwą, krojami pisma i innymi materiałami do budowy interfejsów oraz prześledzić techniki designerskie pomocne w tworzeniu interfejsów nowoczesnych komputerów i skomputeryzowanych urządzeń. Opanowuje praktycznie wybrane narzędzie projektowania makiet interfejsów graficznych.</p>																										
14.	<p>Treści programowe</p> <p>Niżej podano tytuły wykładów. Szczegółowe wypunktowanie poruszanych zagadnień jest sukcesywnie uwidaczniane w trakcie semestru na stronie wykładu w witrynie SKOS.</p> <ol style="list-style-type: none"> 1. Wstęp 2. Wyszukiwanie potrzeb 3. Analiza potrzeb [i] ludzi 4. Prototypowanie czas zacząć 5. Naucz się krytykować 6. Kształtowanie interfejsów 7. Zmysły myślą 8. Barwą, krojem i układem 9. Układ, liczy się układ 10. Inwentaryzacja GUI 11. GUI — dokończenie remanentu 12. Materiały do 17 wykładów studenckich 																										
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 1025 1426 1223"> <tr> <td data-bbox="225 1025 1166 1126">Ma świadomość zasad rządzących konstruowaniem udanych GUI</td> <td data-bbox="1169 1025 1426 1059">K_K08</td> </tr> <tr> <td data-bbox="225 1126 1166 1182">Poznaje miary jakościowe przydatne w ocenie interfejsów (listy kontrolne, w sumie ponad 200 pozycji szczegółowych)</td> <td data-bbox="1169 1126 1426 1160">Inż_W01</td> </tr> <tr> <td data-bbox="225 1182 1166 1238">Opanowuje elementarne, potrzebne w projektowaniu interfejsów wiadomości z zakresu psychologii, socjologii, neurologii, kognitywistyki, typografii i estetyki</td> <td data-bbox="1169 1182 1426 1216">K_W08</td> </tr> <tr> <td data-bbox="225 1238 1166 1272">Zna design i czym zajmują się osoby projektujące interfejsy komputerowe i inne</td> <td data-bbox="1169 1238 1426 1272">K_W08</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 1305 1426 1872"> <tr> <td data-bbox="225 1305 1166 1406">Potrafi uwzględniać w procesie projektowania i kształtowania interfejsów komputerowych zasady działania pamięci człowieka oraz możliwości i ograniczenia zmysłów ludzkich</td> <td data-bbox="1169 1305 1426 1406">K_U10, Inż_U03</td> </tr> <tr> <td data-bbox="225 1406 1166 1496">Potrafi analizować projektowane i gotowe interfejsy, zarówno komputerów, jak i innych skomputeryzowanych przedmiotów i maszyn — także przedmiotów niebędących komputerami</td> <td data-bbox="1169 1406 1426 1496">K_U10, K_U13, Inż_U03</td> </tr> <tr> <td data-bbox="225 1496 1166 1563">Potrafi wykorzystać w projektowaniu interfejsów elementarną wiedzę z zakresu ergonomii</td> <td data-bbox="1169 1496 1426 1563">K_U10, K_U13, Inż_U03</td> </tr> <tr> <td data-bbox="225 1563 1166 1630">Potrafi krytycznie oceniać interfejsy — zarówno komputerowe, jak i inne, dotyczące maszyn, przedmiotów i urządzeń (także codziennego użytku)</td> <td data-bbox="1169 1563 1426 1630">K_U10, K_U13</td> </tr> <tr> <td data-bbox="225 1630 1166 1697">Potrafi sporządzić wykaz potrzeb — wcześniej je dostrzegając (wydobywając) — na podstawie obserwacji, wywiadów, ankiet, list kontrolnych</td> <td data-bbox="1169 1630 1426 1697">K_U13, Inż_U01</td> </tr> <tr> <td data-bbox="225 1697 1166 1731">Potrafi wynajdywać możliwości ulepszeń w istniejących interfejsach</td> <td data-bbox="1169 1697 1426 1731">K_U10, Inż_U05</td> </tr> <tr> <td data-bbox="225 1731 1166 1765">Potrafi testować jakość i przydatność poszczególnych interfejsów</td> <td data-bbox="1169 1731 1426 1765">K_U10, Inż_U05</td> </tr> <tr> <td data-bbox="225 1765 1166 1821">Potrafi operować barwą, odcieniem, kształtem, przestrzenią, proporcjami i rozmieszczeniem elementów interfejsów, "gospodarować pustką" itp</td> <td data-bbox="1169 1765 1426 1821">K_U10, Inż_U02</td> </tr> <tr> <td data-bbox="225 1821 1166 1872">Potrafi w sposób elementarny dobierać kroje pisma i sporządzić projekt aspektów typograficznych interfejsu</td> <td data-bbox="1169 1821 1426 1872">K_U10, Inż_U02</td> </tr> </table>	Ma świadomość zasad rządzących konstruowaniem udanych GUI	K_K08	Poznaje miary jakościowe przydatne w ocenie interfejsów (listy kontrolne, w sumie ponad 200 pozycji szczegółowych)	Inż_W01	Opanowuje elementarne, potrzebne w projektowaniu interfejsów wiadomości z zakresu psychologii, socjologii, neurologii, kognitywistyki, typografii i estetyki	K_W08	Zna design i czym zajmują się osoby projektujące interfejsy komputerowe i inne	K_W08	Potrafi uwzględniać w procesie projektowania i kształtowania interfejsów komputerowych zasady działania pamięci człowieka oraz możliwości i ograniczenia zmysłów ludzkich	K_U10, Inż_U03	Potrafi analizować projektowane i gotowe interfejsy, zarówno komputerów, jak i innych skomputeryzowanych przedmiotów i maszyn — także przedmiotów niebędących komputerami	K_U10, K_U13, Inż_U03	Potrafi wykorzystać w projektowaniu interfejsów elementarną wiedzę z zakresu ergonomii	K_U10, K_U13, Inż_U03	Potrafi krytycznie oceniać interfejsy — zarówno komputerowe, jak i inne, dotyczące maszyn, przedmiotów i urządzeń (także codziennego użytku)	K_U10, K_U13	Potrafi sporządzić wykaz potrzeb — wcześniej je dostrzegając (wydobywając) — na podstawie obserwacji, wywiadów, ankiet, list kontrolnych	K_U13, Inż_U01	Potrafi wynajdywać możliwości ulepszeń w istniejących interfejsach	K_U10, Inż_U05	Potrafi testować jakość i przydatność poszczególnych interfejsów	K_U10, Inż_U05	Potrafi operować barwą, odcieniem, kształtem, przestrzenią, proporcjami i rozmieszczeniem elementów interfejsów, "gospodarować pustką" itp	K_U10, Inż_U02	Potrafi w sposób elementarny dobierać kroje pisma i sporządzić projekt aspektów typograficznych interfejsu	K_U10, Inż_U02
Ma świadomość zasad rządzących konstruowaniem udanych GUI	K_K08																										
Poznaje miary jakościowe przydatne w ocenie interfejsów (listy kontrolne, w sumie ponad 200 pozycji szczegółowych)	Inż_W01																										
Opanowuje elementarne, potrzebne w projektowaniu interfejsów wiadomości z zakresu psychologii, socjologii, neurologii, kognitywistyki, typografii i estetyki	K_W08																										
Zna design i czym zajmują się osoby projektujące interfejsy komputerowe i inne	K_W08																										
Potrafi uwzględniać w procesie projektowania i kształtowania interfejsów komputerowych zasady działania pamięci człowieka oraz możliwości i ograniczenia zmysłów ludzkich	K_U10, Inż_U03																										
Potrafi analizować projektowane i gotowe interfejsy, zarówno komputerów, jak i innych skomputeryzowanych przedmiotów i maszyn — także przedmiotów niebędących komputerami	K_U10, K_U13, Inż_U03																										
Potrafi wykorzystać w projektowaniu interfejsów elementarną wiedzę z zakresu ergonomii	K_U10, K_U13, Inż_U03																										
Potrafi krytycznie oceniać interfejsy — zarówno komputerowe, jak i inne, dotyczące maszyn, przedmiotów i urządzeń (także codziennego użytku)	K_U10, K_U13																										
Potrafi sporządzić wykaz potrzeb — wcześniej je dostrzegając (wydobywając) — na podstawie obserwacji, wywiadów, ankiet, list kontrolnych	K_U13, Inż_U01																										
Potrafi wynajdywać możliwości ulepszeń w istniejących interfejsach	K_U10, Inż_U05																										
Potrafi testować jakość i przydatność poszczególnych interfejsów	K_U10, Inż_U05																										
Potrafi operować barwą, odcieniem, kształtem, przestrzenią, proporcjami i rozmieszczeniem elementów interfejsów, "gospodarować pustką" itp	K_U10, Inż_U02																										
Potrafi w sposób elementarny dobierać kroje pisma i sporządzić projekt aspektów typograficznych interfejsu	K_U10, Inż_U02																										

16.	<p>Literatura obowiązkowa i zalecana</p> <p>Istnieje wiele dobrych podręczników KCK (HCI). Niektóre są dostępne bezpłatnie w Sieci. Prócz tego Sieć zawiera mnóstwo artykułów poświęconych projektowaniu witryn i ogólniej — kwestiom designerskim interfejsów komputerowych. Oto niektóre lektury, oprócz dodatkowych, proponowanych na wykładzie:</p> <ol style="list-style-type: none"> 1. Lynch P. J., Horton S.: Web Style Guide. 3rd Edition. Yale University Press 2008. Książka dostępna w Sieci bez opłat. 2. Van Duyne D. K., Landay J. A., Hong J. I.: The Design of Sites: Patterns for Creating Winning Web Sites. Prentice Hall Professional 2007. 3. Leventhal L. M., Barnes J. A.: Usability Engineering: Process, Products, and Examples. Pearson/Prentice Hall 2008. 4. Dzieła klasyków dziedziny, D. Normana i J. Nielsena – dostępne w Sieci. 5. Inne wartościowe artykuły, przemyślenia, instruktaże i normy, m.in. na stronach firm Microsoft i Google. 6. Kilkadziesiąt zestawów materiałów do wykładów studenckich, dostępnych na stronie wykładu w Sieci. 																		
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <ul style="list-style-type: none"> • Ocena wypowiedzi, pomysłów i uwag osób studiujących, dokonywana podczas ćwiczeń. • Ocena (krytyka) etapów realizacji projektów na pracowni. • Ocena prezentowanych zadań specjalnych i dodatkowych. • Ocena wykładów studenckich, poszerzających i unowocześniających dziedzinę; dla ochotników. • Egzamin pisemny (wybór z ponad 360 pytań). 																		
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu</p> <p>Zaliczenie ćwiczeń i związanej z wykładem pracowni wymaga:</p> <ul style="list-style-type: none"> • aktywnego, twórczego uczestnictwa w ćwiczeniach i pracowni; • opracowania wartościowej i pełnej makiety wybranego interfejsu (w dążeniu do ulepszenia istniejącej aplikacji lub zupełnie nowego); • wykazania się umiejętnością konstruktywnych ocen i krytyki przedmiotów użytkowych, w tym aplikacji komputerowych; • umiejętności przedstawienia i przekazania własnego wyrobu (GUI). <p>Dodatkowo, w miarę możliwości czasowych, studenci mogą przygotować na koniec semestru własne wykłady wzbogacające, a zwłaszcza unowocześniające, wiedzę o trendach, modach i praktyce w dziedzinie KCK (HCI).</p>																		
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 1339 1426 1406"> <tr> <td>ćwiczenie-pracownia</td> <td>30 godz.</td> </tr> <tr> <td>wykład</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 1485 1426 1644"> <tr> <td>przygotowanie do ćwiczeń</td> <td>15 godz.</td> </tr> <tr> <td>praca nad projektem</td> <td>30 godz.</td> </tr> <tr> <td>studiowanie tematyki wykładów i literatury</td> <td>30 godz.</td> </tr> <tr> <td>przygotowanie do pracowni</td> <td>15 godz.</td> </tr> <tr> <td>udział w egzaminie</td> <td>2 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 1727 1426 1796"> <tr> <td>Łączna liczba godzin</td> <td>152 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>6</td> </tr> </table>	ćwiczenie-pracownia	30 godz.	wykład	30 godz.	przygotowanie do ćwiczeń	15 godz.	praca nad projektem	30 godz.	studiowanie tematyki wykładów i literatury	30 godz.	przygotowanie do pracowni	15 godz.	udział w egzaminie	2 godz.	Łączna liczba godzin	152 godz.	Liczba punktów ECTS	6
ćwiczenie-pracownia	30 godz.																		
wykład	30 godz.																		
przygotowanie do ćwiczeń	15 godz.																		
praca nad projektem	30 godz.																		
studiowanie tematyki wykładów i literatury	30 godz.																		
przygotowanie do pracowni	15 godz.																		
udział w egzaminie	2 godz.																		
Łączna liczba godzin	152 godz.																		
Liczba punktów ECTS	6																		

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Logika cyfrowa Digital logic
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-LogCyf
6.	Rodzaj przedmiotu/modułu Informatyczny inżynierski (Iinż)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenio-pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: <ul style="list-style-type: none"> • Logika dla informatyków • Wstęp do informatyki Niezbędne kompetencje: <ul style="list-style-type: none"> • Umiejętność programowania w dowolnym języku programowania • Znajomość podstaw rachunku zdań
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest nauczenie zasad rządzących funkcjonowaniem elektroniki cyfrowej oraz typowych rozwiązań stosowanych przy projektowaniu układów. Zajęcia koncentrują się na następujących pojęciach: logika kombinacyjna i sekwencyjna, algebra Boole'a, automaty skończone. Przedmiot ma charakter praktyczny, jego ważnym elementem jest nauczenie posługiwania się językiem opisu sprzętu.

14.	<p>Treści programowe</p> <ul style="list-style-type: none"> • Wprowadzenie do układów cyfrowych • Technologie implementacji układów cyfrowych • Elementy SystemVeriloga jako języka opisu sprzętu • Optymalizacja funkcji logicznych • Układy arytmetyczne • Podstawowe układy kombinacyjne • Zatraski i przerzutniki • Liczniki • Układy pamięci ROM i RAM • Układy programowalne PAL, PLA, podstawy CPLD i FPGA • Automaty Moore'a i Mealy'ego • Układy sekwencyjne • Podstawy architektury RISC V • Budowa prostego procesora 																		
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%; padding: 5px;">zna podstawowe elementy budowy układów cyfrowych - bramki logiczne - oraz ich związek z algebrą Boole'a</td> <td style="width: 30%; padding: 5px;">K_W08, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">posiada wiedzę na temat reprezentowania danych w formie cyfrowej</td> <td style="padding: 5px;">K_W08, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">zna podstawowe układy bramkowe kombinacyjne i sekwencyjne, w tym: multipleksery, przerzutniki, liczniki, pamięci</td> <td style="padding: 5px;">K_W07, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">zna pojęcie automatu skończonego, automatu Moore'a i Mealy'ego</td> <td style="padding: 5px;">K_W08</td> </tr> <tr> <td style="padding: 5px;">rozumie działanie podstawowych implementacji procesorów (jednocyfrowej, wielocyfrowej i potokowej)</td> <td style="padding: 5px;">K_W07, Inż_W01</td> </tr> </table> <p>Umiejętności</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%; padding: 5px;">potrafi ocenić efektywność (długość ścieżki krytycznej) i rozmiar (liczba bramek) układu cyfrowego</td> <td style="width: 30%; padding: 5px;">K_U09, Inż_U04</td> </tr> <tr> <td style="padding: 5px;">potrafi zaprojektować układ cyfrowy zoptymalizowany ze względu na szybkość, koszt lub zużycie energii</td> <td style="padding: 5px;">K_U07, Inż_U03, Inż_U05</td> </tr> <tr> <td style="padding: 5px;">potrafi projektować i minimalizować automaty skończone oraz implementować z ich pomocą algorytmy jako układy sekwencyjne z wydzieloną ścieżką sterowania i danych</td> <td style="padding: 5px;">K_U07, Inż_U01, Inż_U02</td> </tr> <tr> <td style="padding: 5px;">potrafi modelować układy kombinacyjne i sekwencyjne w języku SystemVerilog</td> <td style="padding: 5px;">K_U03, Inż_U02, Inż_U05</td> </tr> </table>	zna podstawowe elementy budowy układów cyfrowych - bramki logiczne - oraz ich związek z algebrą Boole'a	K_W08, Inż_W01	posiada wiedzę na temat reprezentowania danych w formie cyfrowej	K_W08, Inż_W01	zna podstawowe układy bramkowe kombinacyjne i sekwencyjne, w tym: multipleksery, przerzutniki, liczniki, pamięci	K_W07, Inż_W01	zna pojęcie automatu skończonego, automatu Moore'a i Mealy'ego	K_W08	rozumie działanie podstawowych implementacji procesorów (jednocyfrowej, wielocyfrowej i potokowej)	K_W07, Inż_W01	potrafi ocenić efektywność (długość ścieżki krytycznej) i rozmiar (liczba bramek) układu cyfrowego	K_U09, Inż_U04	potrafi zaprojektować układ cyfrowy zoptymalizowany ze względu na szybkość, koszt lub zużycie energii	K_U07, Inż_U03, Inż_U05	potrafi projektować i minimalizować automaty skończone oraz implementować z ich pomocą algorytmy jako układy sekwencyjne z wydzieloną ścieżką sterowania i danych	K_U07, Inż_U01, Inż_U02	potrafi modelować układy kombinacyjne i sekwencyjne w języku SystemVerilog	K_U03, Inż_U02, Inż_U05
zna podstawowe elementy budowy układów cyfrowych - bramki logiczne - oraz ich związek z algebrą Boole'a	K_W08, Inż_W01																		
posiada wiedzę na temat reprezentowania danych w formie cyfrowej	K_W08, Inż_W01																		
zna podstawowe układy bramkowe kombinacyjne i sekwencyjne, w tym: multipleksery, przerzutniki, liczniki, pamięci	K_W07, Inż_W01																		
zna pojęcie automatu skończonego, automatu Moore'a i Mealy'ego	K_W08																		
rozumie działanie podstawowych implementacji procesorów (jednocyfrowej, wielocyfrowej i potokowej)	K_W07, Inż_W01																		
potrafi ocenić efektywność (długość ścieżki krytycznej) i rozmiar (liczba bramek) układu cyfrowego	K_U09, Inż_U04																		
potrafi zaprojektować układ cyfrowy zoptymalizowany ze względu na szybkość, koszt lub zużycie energii	K_U07, Inż_U03, Inż_U05																		
potrafi projektować i minimalizować automaty skończone oraz implementować z ich pomocą algorytmy jako układy sekwencyjne z wydzieloną ścieżką sterowania i danych	K_U07, Inż_U01, Inż_U02																		
potrafi modelować układy kombinacyjne i sekwencyjne w języku SystemVerilog	K_U03, Inż_U02, Inż_U05																		
16.	<p>Literatura obowiązkowa i zalecana</p> <p>Literatura obowiązkowa:</p> <ul style="list-style-type: none"> • Fundamentals of Digital Logic with Verilog Design; Brown, Vranesic; McGraw-Hill [roz. 1-7] • Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog (6th Edition); Mano, Ciletti; Pearson • Computer Organization and Design, the Hardware/Software Interface, RISC-V Edition [roz. 4] <p>Literatura zalecana:</p> <ul style="list-style-type: none"> • Verilog HDL (2nd Edition); Palnitkar • A Verilog HDL Primer, Third Edition; Bhasker • Standard IEEE 1364-2005 (Verilog 2005) • Standard IEEE 1800-2012 (SystemVerilog 2012) • Lessons in Electric Circuits; Kuphaldt [tom IV] 																		
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>prezentacja rozwiązania zadania, napisanie programu w języku opisu sprzętu, egzamin pisemny</p>																		

18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Do zaliczenia ćwiczenio-pracowni należy zdobyć wymaganą, podaną w regulaminie przedmiotu liczbę punktów za zadania ćwiczeniowe i pracowniowe. Końcowa punktacja jest liczona przy użyciu średniej ważonej, aby waga ćwiczeń i pracowni wynosiła 50. Zadania ćwiczeniowe sprawdzane są podczas prezentacji przy tablicy. Na punktację zadań pracowniowych składa się wyliczana automatycznie ocena za poprawność oraz indywidualna ocena prowadzącego.</p> <p>Egzamin ma formę pisemną, do jego zaliczenia konieczne jest zdobycie odpowiedniej liczby punktów.</p>																		
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 443 1428 510"> <tr> <td data-bbox="225 443 1166 472">ćwiczenio-pracownia</td> <td data-bbox="1166 443 1428 472">30 godz.</td> </tr> <tr> <td data-bbox="225 472 1166 510">wykład</td> <td data-bbox="1166 472 1428 510">30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 584 1428 745"> <tr> <td data-bbox="225 584 1166 613">studiowanie tematyki wykładów i literatury</td> <td data-bbox="1166 584 1428 613">20 godz.</td> </tr> <tr> <td data-bbox="225 613 1166 642">przygotowanie do pracowni</td> <td data-bbox="1166 613 1428 642">30 godz.</td> </tr> <tr> <td data-bbox="225 642 1166 672">przygotowanie do ćwiczeń</td> <td data-bbox="1166 642 1428 672">30 godz.</td> </tr> <tr> <td data-bbox="225 672 1166 701">przygotowanie do egzaminu</td> <td data-bbox="1166 672 1428 701">7 godz.</td> </tr> <tr> <td data-bbox="225 701 1166 745">udział w egzaminie</td> <td data-bbox="1166 701 1428 745">3 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 831 1428 898"> <tr> <td data-bbox="225 831 1166 860">Łączna liczba godzin</td> <td data-bbox="1166 831 1428 860">150 godz.</td> </tr> <tr> <td data-bbox="225 860 1166 898">Liczba punktów ECTS</td> <td data-bbox="1166 860 1428 898">6</td> </tr> </table>	ćwiczenio-pracownia	30 godz.	wykład	30 godz.	studiowanie tematyki wykładów i literatury	20 godz.	przygotowanie do pracowni	30 godz.	przygotowanie do ćwiczeń	30 godz.	przygotowanie do egzaminu	7 godz.	udział w egzaminie	3 godz.	Łączna liczba godzin	150 godz.	Liczba punktów ECTS	6
ćwiczenio-pracownia	30 godz.																		
wykład	30 godz.																		
studiowanie tematyki wykładów i literatury	20 godz.																		
przygotowanie do pracowni	30 godz.																		
przygotowanie do ćwiczeń	30 godz.																		
przygotowanie do egzaminu	7 godz.																		
udział w egzaminie	3 godz.																		
Łączna liczba godzin	150 godz.																		
Liczba punktów ECTS	6																		

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Podstawy grafiki komputerowej Fundamentals of Computer Graphics
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DP6PGK
6.	Rodzaj przedmiotu/modułu Informatyczny inżynierski (Iinż)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Niezbędne kompetencje: <ul style="list-style-type: none"> • Znajomość algebry liniowej, analizy numerycznej, podstaw algorytmów i struktur danych. • Umiejętność dobrego programowania w C/C++.
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest przedstawienie podstawowych pojęć i algorytmów grafiki komputerowej oraz nabycie przez studentów umiejętności programowania kart graficznych. Celem pracowni jest nauka praktyki programowania w nowym OpenGL.
14.	Treści programowe <ul style="list-style-type: none"> • Programowanie kart graficznych przy pomocy API OpenGL • Przekształcenia w jednorodnym układzie współrzędnych • Efektywne reprezentacje rotacji w tym kwaterniony • Modelowanie wirtualnej kamery i obiektów 3D • Widoczność: algorytm z-bufora, śledzenie promieni i inne • Modelowanie i obliczanie oświetlenia • Rasteryzacja i antialiasing • Światło i kolor w grafice, modele koloru • Teksturowanie 2D/3D, tekstury proceduralne

15.	Zakładane efekty uczenia się	
	Wiedza	
	zna podstawowe pojęcia i algorytmy grafiki komputerowej rozumie ograniczenia i podstawy działania GPU	K_W08, Inż_W01 K_W07, Inż_W01
	Umiejętności	
umie konstruować macierze w jednorodnym układzie współrzędnych dla przekształceń pomiędzy zadanymi układami potrafi programować karty graficzne przy pomocy API OpenGL	K_U07, Inż_U02, K_U03, Inż_U01, Inż_U04, Inż_U05	
umie programować shadery w języku GLSL	K_U03, Inż_U01, Inż_U03, Inż_U04, Inż_U05	
Kompetencje społeczne		
rozumie znaczenie grafiki komputerowej we współczesnym społeczeństwie	K_K05 K_K06	
16.	Literatura obowiązkowa i zalecana <ul style="list-style-type: none"> • P. Shirley - "Fundamentals of Computer Graphics", A.K.Peters, Natick Massachusetts 2002. • J.F.Hughes, A.van Dam, M.Mcguire, D.F.Sklar, J.D.Foley, S.K.Feiner, K.Akeley - "Computer Graphics - Principles and Practice", 3rd Edition 2014. 	
17.	Metody weryfikacji zakładanych efektów uczenia się <ul style="list-style-type: none"> • napisanie programu komputerowego • prezentacja rozwiązania zadania • egzamin 	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia pracowni należy zdobyć wymaganą, podaną w regulaminie przedmiotu, liczbę punktów za zadania/projekty programistyczne. Aby zaliczyć egzamin konieczne jest zdobycie wymaganej liczby punktów.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
	przygotowanie do egzaminu	10 godz.
	studiowanie tematyki wykładów i literatury	30 godz.
przygotowanie do pracowni	20 godz.	
egzamin	2 godz.	
praca nad projektami programistycznymi	30 godz.	
Sumarycznie		
Łączna liczba godzin	152 godz.	
Liczba punktów ECTS	6	

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projektowanie obiektowe oprogramowania Object-oriented Software Development
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DZ6POO
6.	Rodzaj przedmiotu/modułu Informatyczny inżynierski (Iinż) ze znacznikami IO oraz PiPO
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 30 godzin, pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Niezbędne kompetencje <ul style="list-style-type: none"> • znajomość języka obiektowego z grupy C/Java/Scala
13.	Cele przedmiotu Wykład skierowany jest do przyszłych architektów i projektantów systemów informatycznych oraz do wszystkich programistów zainteresowanych udoskonaleniem swojego warsztatu. Celem wykładu jest zapoznanie studentów z kanonem współczesnych narzędzi w zakresie projektowania obiektowego oprogramowania.
14.	Treści programowe <ul style="list-style-type: none"> • Projektowanie obiektowe: zbieranie wymagań, ramy FURPS i SMART • Język UML • Zasady SOLID i GRASP • Wzorce projektowe GoF • Wzorce architektury aplikacji (w tym DI, ORM, MVP, ESB, CQRS) • Testowanie, ramy typów zastępczych

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="228 192 1157 383"> <tr> <td>zna zasady projektowania obiektowego</td> <td>K_W05, Inż_W01</td> </tr> <tr> <td>zna wybrane wzorce projektowe</td> <td>K_W07, Inż_W01</td> </tr> <tr> <td>zna wybrane wzorce architektury aplikacji</td> <td>K_W07, Inż_W01</td> </tr> <tr> <td>zna język UML</td> <td>K_W07, Inż_W01</td> </tr> <tr> <td>zna wybrane metodyki wytwarzania aplikacji</td> <td>K_W07, Inż_W01</td> </tr> <tr> <td>zna sposoby testowania aplikacji</td> <td>K_W07, Inż_W01</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="228 468 1157 658"> <tr> <td>potrafi zbudować dokumentację analityczną systemu przy pomocy wybranego narzędzia do projektowania UML</td> <td>K_U07, K_U13, Inż_U05</td> </tr> <tr> <td>potrafi projektować architekturę systemu w zależności od wymagań</td> <td>K_U08, Inż_U02, Inż_U03</td> </tr> <tr> <td>potrafi implementować poszczególne elementy stosu aplikacyjnego zgodnie z praktyką przemysłową</td> <td>K_U03, Inż_U01, Inż_U04</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="228 743 1157 804"> <tr> <td>rozumie konieczność ciągłego dokształcania się w obliczu zmieniających się technologii</td> <td>K_K01, K_K06</td> </tr> </table>	zna zasady projektowania obiektowego	K_W05, Inż_W01	zna wybrane wzorce projektowe	K_W07, Inż_W01	zna wybrane wzorce architektury aplikacji	K_W07, Inż_W01	zna język UML	K_W07, Inż_W01	zna wybrane metodyki wytwarzania aplikacji	K_W07, Inż_W01	zna sposoby testowania aplikacji	K_W07, Inż_W01	potrafi zbudować dokumentację analityczną systemu przy pomocy wybranego narzędzia do projektowania UML	K_U07, K_U13, Inż_U05	potrafi projektować architekturę systemu w zależności od wymagań	K_U08, Inż_U02, Inż_U03	potrafi implementować poszczególne elementy stosu aplikacyjnego zgodnie z praktyką przemysłową	K_U03, Inż_U01, Inż_U04	rozumie konieczność ciągłego dokształcania się w obliczu zmieniających się technologii	K_K01, K_K06	
zna zasady projektowania obiektowego	K_W05, Inż_W01																					
zna wybrane wzorce projektowe	K_W07, Inż_W01																					
zna wybrane wzorce architektury aplikacji	K_W07, Inż_W01																					
zna język UML	K_W07, Inż_W01																					
zna wybrane metodyki wytwarzania aplikacji	K_W07, Inż_W01																					
zna sposoby testowania aplikacji	K_W07, Inż_W01																					
potrafi zbudować dokumentację analityczną systemu przy pomocy wybranego narzędzia do projektowania UML	K_U07, K_U13, Inż_U05																					
potrafi projektować architekturę systemu w zależności od wymagań	K_U08, Inż_U02, Inż_U03																					
potrafi implementować poszczególne elementy stosu aplikacyjnego zgodnie z praktyką przemysłową	K_U03, Inż_U01, Inż_U04																					
rozumie konieczność ciągłego dokształcania się w obliczu zmieniających się technologii	K_K01, K_K06																					
16.	<p>Literatura obowiązkowa i zalecana</p> <ol style="list-style-type: none"> 1. Wrycza, Marcinkowski, Wyrzykowski - Język UML 2.0 w modelowaniu systemów informatycznych 2. Fowler - Refactoring: Improving the Design of Existing Code 3. Gamma, Helm, Johnson, Vlissides: Design Patterns: Elements of Reusable Object- Oriented Software 4. B.Martin, M.Martin. Programowanie zwinne: zasady, wzorce i praktyki zwinnego wytwarzania oprogramowania w C#. 5. Larman - UML i wzorce projektowe. Analiza i projektowanie obiektowe oraz iteracyjny model wytwarzania aplikacji 6. Fowler - Patterns of Enterprise Application Architecture 7. Microsoft Patterns Practices - Application Architecture Guide 																					
17.	<p>Metody weryfikacji zakładanych efektów kształcenia</p> <ul style="list-style-type: none"> • napisanie i prezentacja programu komputerowego • prezentacja rozwiązania zadania • egzamin pisemny 																					
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Wykład, pracownia programistyczna - listy zadań pokrywające poszczególne partie materiału. Student zdobywa minimalną liczbę punktów określoną w regulaminie zajęć.</p> <p>Egzamin pisemny sprawdzający wiedzę w poszczególnych partiach materiału.</p>																					
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="228 1556 1157 1617"> <tr> <td>pracownia</td> <td>30 godz.</td> </tr> <tr> <td>wykład</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="228 1702 1157 1827"> <tr> <td>udział w egzaminie</td> <td>2 godz.</td> </tr> <tr> <td>przygotowanie do pracowni</td> <td>40 godz.</td> </tr> <tr> <td>studiowanie tematyki wykładów i literatury</td> <td>20 godz.</td> </tr> <tr> <td>przygotowanie do egzaminu</td> <td>20 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="228 1912 1157 1973"> <tr> <td>Łączna liczba godzin</td> <td>142 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>6</td> </tr> </table>	pracownia	30 godz.	wykład	30 godz.	udział w egzaminie	2 godz.	przygotowanie do pracowni	40 godz.	studiowanie tematyki wykładów i literatury	20 godz.	przygotowanie do egzaminu	20 godz.	Łączna liczba godzin	142 godz.	Liczba punktów ECTS	6					
pracownia	30 godz.																					
wykład	30 godz.																					
udział w egzaminie	2 godz.																					
przygotowanie do pracowni	40 godz.																					
studiowanie tematyki wykładów i literatury	20 godz.																					
przygotowanie do egzaminu	20 godz.																					
Łączna liczba godzin	142 godz.																					
Liczba punktów ECTS	6																					

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Systemy wbudowane Embedded systems
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DP11SW
6.	Rodzaj przedmiotu/modułu Informatyczny inżynierski (Iinż) ze znacznikiem ASK
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: <ul style="list-style-type: none"> • Wstęp do informatyki • Wskazane: systemy operacyjne • Wskazane: logika cyfrowa Niezbędne kompetencje: <ul style="list-style-type: none"> • Umiejętność programowania w języku C • Wskazane: znajomość podstaw elektroniki
13.	Cele kształcenia dla przedmiotu Celem wykładu jest nauczenie podstaw programowania mikrokontrolerów oraz projektowania układów elektronicznych z mikrokontrolerami. Na zajęciach student pozna, w jaki sposób wykorzystać peryferia mikrokontrolerów do sterowania różnorodnymi urządzeniami zewnętrznymi.

14.	<p>Treści programowe</p> <ul style="list-style-type: none"> • Wprowadzenie do systemów wbudowanych • Podstawy architektury AVR • Podstawy systemu FreeRTOS • Prototypowanie układów na płytce stykowej • Projektowanie układów elektronicznych z mikrokontrolerem • Obsługa portów GPIO mikrokontrolera • Przetworniki analogowo-cyfrowe • Fala prostokątna, modulacja PWM • Obsługa przerwań na mikrokontrolerach • Zarządzanie energią • Protokoły komunikacyjne UART, SPI, I2C • Elementy przełączające • Podstawy wzmacniaczy • Sterowanie silnikami • Podstawy teorii sterowania 														
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 719 1428 882"> <tr> <td>Rozumie na podstawowym poziomie działanie najważniejszych rodzajów elementów elektronicznych</td> <td>K_W07</td> </tr> <tr> <td>Posiada przeglądową wiedzę na temat obwodów peryferyjnych występujących w popularnych mikrokontrolerach: liczników, generatorów PWM, transceiverów, przetworników ADC</td> <td>K_W08, Inż_W01</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 965 1428 1256"> <tr> <td>Potrafi konstruować prototypy układów elektronicznych z mikrokontrolerem na podstawie schematu przy użyciu płytki prototypowej</td> <td>K_U9, Inż_U01, Inż_U04, Inż_U05</td> </tr> <tr> <td>Potrafi programować mikrokontrolery AVR ATmega przy użyciu języka C oraz wykorzystywać system czasu rzeczywistego FreeRTOS</td> <td>K_U3, Inż_U01</td> </tr> <tr> <td>Potrafi przeczytać notę katalogową układu elektronicznego i na jej podstawie zaimplementować sterownik tego układu</td> <td>K_U7, Inż_U1, Inż_U3</td> </tr> <tr> <td>Potrafi dobrać właściwą metodę komunikacji (np. protokół UART, SPI lub I2C, sygnał analogowy, sygnał PWM) pomiędzy mikrokontrolerem a urządzeniem lub innym mikrokontrolerem</td> <td>K_U9, Inż_U1, Inż_U2, Inż_U4</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 1339 1428 1406"> <tr> <td>Rozumie znaczenie konwencji stosowanych w schematach elektronicznych dla komunikacji międzyludzkiej</td> <td>K_K02, K_K06</td> </tr> </table>	Rozumie na podstawowym poziomie działanie najważniejszych rodzajów elementów elektronicznych	K_W07	Posiada przeglądową wiedzę na temat obwodów peryferyjnych występujących w popularnych mikrokontrolerach: liczników, generatorów PWM, transceiverów, przetworników ADC	K_W08, Inż_W01	Potrafi konstruować prototypy układów elektronicznych z mikrokontrolerem na podstawie schematu przy użyciu płytki prototypowej	K_U9, Inż_U01, Inż_U04, Inż_U05	Potrafi programować mikrokontrolery AVR ATmega przy użyciu języka C oraz wykorzystywać system czasu rzeczywistego FreeRTOS	K_U3, Inż_U01	Potrafi przeczytać notę katalogową układu elektronicznego i na jej podstawie zaimplementować sterownik tego układu	K_U7, Inż_U1, Inż_U3	Potrafi dobrać właściwą metodę komunikacji (np. protokół UART, SPI lub I2C, sygnał analogowy, sygnał PWM) pomiędzy mikrokontrolerem a urządzeniem lub innym mikrokontrolerem	K_U9, Inż_U1, Inż_U2, Inż_U4	Rozumie znaczenie konwencji stosowanych w schematach elektronicznych dla komunikacji międzyludzkiej	K_K02, K_K06
Rozumie na podstawowym poziomie działanie najważniejszych rodzajów elementów elektronicznych	K_W07														
Posiada przeglądową wiedzę na temat obwodów peryferyjnych występujących w popularnych mikrokontrolerach: liczników, generatorów PWM, transceiverów, przetworników ADC	K_W08, Inż_W01														
Potrafi konstruować prototypy układów elektronicznych z mikrokontrolerem na podstawie schematu przy użyciu płytki prototypowej	K_U9, Inż_U01, Inż_U04, Inż_U05														
Potrafi programować mikrokontrolery AVR ATmega przy użyciu języka C oraz wykorzystywać system czasu rzeczywistego FreeRTOS	K_U3, Inż_U01														
Potrafi przeczytać notę katalogową układu elektronicznego i na jej podstawie zaimplementować sterownik tego układu	K_U7, Inż_U1, Inż_U3														
Potrafi dobrać właściwą metodę komunikacji (np. protokół UART, SPI lub I2C, sygnał analogowy, sygnał PWM) pomiędzy mikrokontrolerem a urządzeniem lub innym mikrokontrolerem	K_U9, Inż_U1, Inż_U2, Inż_U4														
Rozumie znaczenie konwencji stosowanych w schematach elektronicznych dla komunikacji międzyludzkiej	K_K02, K_K06														
16.	<p>Literatura obowiązkowa i zalecana</p> <p>Literatura obowiązkowa:</p> <ul style="list-style-type: none"> • Make: AVR programming; Elliot Williams; Maker Media, Inc. • Designing Embedded Hardware; John Catsoulis; O'Reilly [rozdziały 4, 6-9, 13, 15] <p>Literatura zalecana:</p> <ul style="list-style-type: none"> • Lessons in Electric Circuits; Tony R. Kuphaldt 														
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>budowa prototypu układu elektronicznego z mikrokontrolerem, oprogramowanie mikrokontrolera, egzamin pisemny</p>														
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Do zaliczenia pracowni należy zdobyć wymaganą, podaną w regulaminie przedmiotu liczbę punktów za zadania. Zadania są rozwiązywane w domu, ich rozwiązania (złożone z prototypu układu i programu na mikrokontroler) prezentowane są na pracowni.</p> <p>Egzamin ma formę pisemną, do jego zaliczenia konieczne jest zdobycie odpowiedniej liczby punktów. Osoby, które osiągnęły bardzo dobre wyniki na ćwiczeniach mogą uzyskać zwolnienie z egzaminu.</p>														

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
	studiowanie tematyki wykładów i literatury	30 godz.
	przygotowanie do pracowni	50 godz.
	udział w egzaminie	3 godz.
	przygotowanie do egzaminu	7 godz.
	Sumarycznie	
Łączna liczba godzin	150 godz.	
Liczba punktów ECTS	6	

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Sztuczna inteligencja Artificial Intelligence
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DP6SI
6.	Rodzaj przedmiotu/modułu Informatyczny inżynierski (Iinż)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: <ul style="list-style-type: none"> • Logika dla informatyków • Wstęp do informatyki lub Algorytmy i struktury danych • Analiza matematyczna Niezbędne kompetencje: <ul style="list-style-type: none"> • Umiejętność programowania w języku wyższego poziomu • Pożądana wstępna znajomość języka Python
13.	Cele kształcenia dla przedmiotu Podstawowym celem przedmiotu jest zapoznanie studentów z technikami stosowanymi do rozwiązywania problemów, które są z jednej strony trudne do rozwiązania przy użyciu standardowych technik algorytmicznych, a z drugiej są efektywnie rozwiązywane przez ludzi, korzystających ze swojej inteligencji. Zajęcia koncentrują się na następujących pojęciach: modelowanie świata, przeszukiwanie przestrzeni rozwiązań, wnioskowanie i uczenie się z przykładów bądź z symulacji.

14.	<p>Treści programowe</p> <ol style="list-style-type: none"> 1. Modelowanie rzeczywistości za pomocą przestrzeni stanów. 2. Przeszukiwanie w przestrzeni stanów: przeszukiwanie wszerek i w głąb, iteracyjne pogłębianie, przeszukiwanie dwustronne, algorytm A*, właściwości i tworzenie funkcji heurystycznych wspomagających przeszukiwanie. 3. Przeszukiwanie metaheurystyczne: hill climbing, symulowane wyżarzanie, beam search, algorytmy ewolucyjne. 4. Rozwiązywanie więzów: modelowanie za pomocą więzów, spójność więzów i algorytm AC-3, łączenie propagacji więzów z przeszukiwaniem z nawrotami, specjalistyczne języki programowania z więzami na przykładzie Prologa z więzami. 5. Strategie w grach: gry dwuosobowe, algorytm minimax, odcięcia alfa-beta, przykłady heurystycznej oceny sytuacji na planszy w wybranych grach, losowość w grach, algorytm Monte Carlo Tree Search. 6. Elementy teorii gier: strategie czyste i mieszane, rozwiązywanie prostych gier typu dylemat więźnia. 7. Modelowanie za pomocą logiki zdaniowej, wnioskowanie forward-chaining i backward-chaining, rozwiązywanie problemów spełnialności formuły w CNF (WalkSAT, DPLL). 8. Modelowanie niepewności świata: sieci bayesowskie, procesy decyzyjne Markowa, algorytmy value iteration oraz policy iteration. Elementy uczenia ze wzmocnieniem: TD-learning oraz Q-learning. 9. Podstawy uczenia maszynowego: idea uczenia się z przykładów, generalizacja, niebezpieczeństwo przeuczenia. Wybrane metody: regresja liniowa i logistyczna, wielowarstwowe sieci neuronowe (MLP), algorytm k-NN, drzewa decyzyjne i lasy losowe. 																								
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 913 1426 1205"> <tr> <td>rozumie, czym zajmuje się Sztuczna inteligencja, rozumie również, na czym polega specyficzność metod tej dziedziny</td> <td>K_W08</td> </tr> <tr> <td>posiada przeglądową wiedzę o różnych dziedzinach sztucznej inteligencji</td> <td>K_W08</td> </tr> <tr> <td>zna różne metody modelowania świata, z uwzględnieniem niepewności</td> <td>K_W08, Inż_W01</td> </tr> <tr> <td>zna algorytmy przeszukiwania przestrzeni stanów oraz przeszukiwania drzew gry</td> <td>K_W08, Inż_W01</td> </tr> <tr> <td>zna podstawowe algorytmy wnioskowania</td> <td>K_W08, Inż_W01</td> </tr> <tr> <td>zna podstawowe metody uczenia maszynowego (z nadzorem oraz ze wzmocnieniem)</td> <td>K_W08, Inż_W01</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 1285 1426 1541"> <tr> <td>umie modelować różne zagadnienia jako zadania przeszukiwania (lub przeszukiwania z więzami)</td> <td>K_U09, Inż_U02, Inż_U03</td> </tr> <tr> <td>umie stosować i modyfikować różne algorytmy przeszukiwania (w tym również przeszukiwania w grach)</td> <td>K_U07, Inż_U04</td> </tr> <tr> <td>umie modelować niepewność świata za pomocą narzędzi z rachunku prawdopodobieństwa (ze szczególnym uwzględnieniem metod Monte Carlo)</td> <td>K_U07, Inż_U01</td> </tr> <tr> <td>umie stosować podstawowe metody uczenia maszynowego (w tym również metody uczenia ze wzmocnieniem)</td> <td>K_U03, Inż_U05</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 1621 1426 1756"> <tr> <td>rozumie znaczenie algorytmów sztucznej inteligencji dla funkcjonowania współczesnego społeczeństwa, rozumie możliwości i niebezpieczeństwa z tym związane</td> <td>K_K02, K_K06</td> </tr> <tr> <td>umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy</td> <td>K_K03, K_K06</td> </tr> </table>	rozumie, czym zajmuje się Sztuczna inteligencja, rozumie również, na czym polega specyficzność metod tej dziedziny	K_W08	posiada przeglądową wiedzę o różnych dziedzinach sztucznej inteligencji	K_W08	zna różne metody modelowania świata, z uwzględnieniem niepewności	K_W08, Inż_W01	zna algorytmy przeszukiwania przestrzeni stanów oraz przeszukiwania drzew gry	K_W08, Inż_W01	zna podstawowe algorytmy wnioskowania	K_W08, Inż_W01	zna podstawowe metody uczenia maszynowego (z nadzorem oraz ze wzmocnieniem)	K_W08, Inż_W01	umie modelować różne zagadnienia jako zadania przeszukiwania (lub przeszukiwania z więzami)	K_U09, Inż_U02, Inż_U03	umie stosować i modyfikować różne algorytmy przeszukiwania (w tym również przeszukiwania w grach)	K_U07, Inż_U04	umie modelować niepewność świata za pomocą narzędzi z rachunku prawdopodobieństwa (ze szczególnym uwzględnieniem metod Monte Carlo)	K_U07, Inż_U01	umie stosować podstawowe metody uczenia maszynowego (w tym również metody uczenia ze wzmocnieniem)	K_U03, Inż_U05	rozumie znaczenie algorytmów sztucznej inteligencji dla funkcjonowania współczesnego społeczeństwa, rozumie możliwości i niebezpieczeństwa z tym związane	K_K02, K_K06	umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy	K_K03, K_K06
rozumie, czym zajmuje się Sztuczna inteligencja, rozumie również, na czym polega specyficzność metod tej dziedziny	K_W08																								
posiada przeglądową wiedzę o różnych dziedzinach sztucznej inteligencji	K_W08																								
zna różne metody modelowania świata, z uwzględnieniem niepewności	K_W08, Inż_W01																								
zna algorytmy przeszukiwania przestrzeni stanów oraz przeszukiwania drzew gry	K_W08, Inż_W01																								
zna podstawowe algorytmy wnioskowania	K_W08, Inż_W01																								
zna podstawowe metody uczenia maszynowego (z nadzorem oraz ze wzmocnieniem)	K_W08, Inż_W01																								
umie modelować różne zagadnienia jako zadania przeszukiwania (lub przeszukiwania z więzami)	K_U09, Inż_U02, Inż_U03																								
umie stosować i modyfikować różne algorytmy przeszukiwania (w tym również przeszukiwania w grach)	K_U07, Inż_U04																								
umie modelować niepewność świata za pomocą narzędzi z rachunku prawdopodobieństwa (ze szczególnym uwzględnieniem metod Monte Carlo)	K_U07, Inż_U01																								
umie stosować podstawowe metody uczenia maszynowego (w tym również metody uczenia ze wzmocnieniem)	K_U03, Inż_U05																								
rozumie znaczenie algorytmów sztucznej inteligencji dla funkcjonowania współczesnego społeczeństwa, rozumie możliwości i niebezpieczeństwa z tym związane	K_K02, K_K06																								
umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy	K_K03, K_K06																								
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach. • Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction. • Prateek Joshi, Artificial Intelligence with Python. 																								
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>egzamin pisemny, prezentacja projektu, prezentacja rozwiązania zadania, napisanie i prezentacja programu komputerowego</p>																								

18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Do zaliczenia ćwiczenio-pracowni należy zdobyć wymaganą, podaną w regulaminie przedmiotu liczbę punktów za zadania ćwiczeniowe, pracowniowe i opcjonalny projekt. Punkty za wszystkie wyżej wymienione aktywności liczą się łącznie. Egzamin ma formę pisemną, aby go zaliczyć konieczne jest zdobycie wymaganej liczby punktów.</p>																
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 376 1428 443"> <tr> <td data-bbox="225 376 1166 409">pracownia</td> <td data-bbox="1166 376 1428 409">30 godz.</td> </tr> <tr> <td data-bbox="225 409 1166 443">wykład</td> <td data-bbox="1166 409 1428 443">30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 521 1428 651"> <tr> <td data-bbox="225 521 1166 555">udział w egzaminie</td> <td data-bbox="1166 521 1428 555">3 godz.</td> </tr> <tr> <td data-bbox="225 555 1166 589">samodzielne rozwiązywanie zadań pracowniowych i projektowych</td> <td data-bbox="1166 555 1428 589">50 godz.</td> </tr> <tr> <td data-bbox="225 589 1166 622">studiowanie tematyki wykładów i literatury oraz przygotowanie do egzaminu</td> <td data-bbox="1166 589 1428 622">15 godz.</td> </tr> <tr> <td data-bbox="225 622 1166 651">przygotowywanie się do ćwiczeń (w tym czytanie materiałów dodatkowych)</td> <td data-bbox="1166 622 1428 651">25 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 734 1428 801"> <tr> <td data-bbox="225 734 1166 768">Łączna liczba godzin</td> <td data-bbox="1166 734 1428 768">153 godz.</td> </tr> <tr> <td data-bbox="225 768 1166 801">Liczba punktów ECTS</td> <td data-bbox="1166 768 1428 801">6</td> </tr> </table>	pracownia	30 godz.	wykład	30 godz.	udział w egzaminie	3 godz.	samodzielne rozwiązywanie zadań pracowniowych i projektowych	50 godz.	studiowanie tematyki wykładów i literatury oraz przygotowanie do egzaminu	15 godz.	przygotowywanie się do ćwiczeń (w tym czytanie materiałów dodatkowych)	25 godz.	Łączna liczba godzin	153 godz.	Liczba punktów ECTS	6
pracownia	30 godz.																
wykład	30 godz.																
udział w egzaminie	3 godz.																
samodzielne rozwiązywanie zadań pracowniowych i projektowych	50 godz.																
studiowanie tematyki wykładów i literatury oraz przygotowanie do egzaminu	15 godz.																
przygotowywanie się do ćwiczeń (w tym czytanie materiałów dodatkowych)	25 godz.																
Łączna liczba godzin	153 godz.																
Liczba punktów ECTS	6																

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Wybrane elementy praktyki projektowania oprogramowania Selected Aspects of Practical Software Development
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DI1WEPP0
6.	Rodzaj przedmiotu/modułu Informatyczny podstawowy (Iinż) ze znacznikami IO oraz PiPO
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu
13.	Cele kształcenia dla przedmiotu Celem zajęć jest przedstawienie studentom praktyki projektowania oprogramowania. Zajęcia w związku z tym łączą podstawową wiedzę z zakresu baz danych i inżynierii oprogramowania w obszarze projektowania obiektowego w wybranych współczesnych realiach technologicznych.
14.	Treści programowe 1. podstawy projektowania obiektowego 2. język UML 3. język Javascript - programowanie modularne i obiektowe 4. język Typescript 5. technologia node.js 6. frameworki Express i Socket.IO 7. podstawy języka SQL i komunikacji z relacyjnymi bazami danych 8. testowanie aplikacji webowych

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="228 192 1157 353"> <tr> <td>zna podstawy projektowania obiektowego</td> <td>K_W07, Inż_W01</td> </tr> <tr> <td>zna podstawy technologii baz danych</td> <td>K_W07, Inż_W01</td> </tr> <tr> <td>zna wybrany przemysłowy język wytwarzania aplikacji</td> <td>K_W07, Inż_W01</td> </tr> <tr> <td>zna możliwości wybranej przemysłowej technologii wytwarzania aplikacji (w tym aplikacji internetowych)</td> <td>K_W07, Inż_W01</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="228 439 1157 689"> <tr> <td>potrafi zaprojektować strukturę relacyjnej bazy danych w wybranej technologii</td> <td>K_U08, Inż_U04, Inż_U05</td> </tr> <tr> <td>potrafi zaprojektować, zaimplementować i wdrożyć aplikację internetową wykorzystującą typowe elementy technologiczne</td> <td>K_U03, K_U07, K_U08, Inż_U01, Inż_U03, Inż_U04, Inż_U05</td> </tr> <tr> <td>potrafi projektować diagramy UML z wykorzystaniem wybranego narzędzia przemysłowego</td> <td>K_U08, Inż_U02, Inż_U04, Inż_U05</td> </tr> </table>	zna podstawy projektowania obiektowego	K_W07, Inż_W01	zna podstawy technologii baz danych	K_W07, Inż_W01	zna wybrany przemysłowy język wytwarzania aplikacji	K_W07, Inż_W01	zna możliwości wybranej przemysłowej technologii wytwarzania aplikacji (w tym aplikacji internetowych)	K_W07, Inż_W01	potrafi zaprojektować strukturę relacyjnej bazy danych w wybranej technologii	K_U08, Inż_U04, Inż_U05	potrafi zaprojektować, zaimplementować i wdrożyć aplikację internetową wykorzystującą typowe elementy technologiczne	K_U03, K_U07, K_U08, Inż_U01, Inż_U03, Inż_U04, Inż_U05	potrafi projektować diagramy UML z wykorzystaniem wybranego narzędzia przemysłowego	K_U08, Inż_U02, Inż_U04, Inż_U05		
zna podstawy projektowania obiektowego	K_W07, Inż_W01																
zna podstawy technologii baz danych	K_W07, Inż_W01																
zna wybrany przemysłowy język wytwarzania aplikacji	K_W07, Inż_W01																
zna możliwości wybranej przemysłowej technologii wytwarzania aplikacji (w tym aplikacji internetowych)	K_W07, Inż_W01																
potrafi zaprojektować strukturę relacyjnej bazy danych w wybranej technologii	K_U08, Inż_U04, Inż_U05																
potrafi zaprojektować, zaimplementować i wdrożyć aplikację internetową wykorzystującą typowe elementy technologiczne	K_U03, K_U07, K_U08, Inż_U01, Inż_U03, Inż_U04, Inż_U05																
potrafi projektować diagramy UML z wykorzystaniem wybranego narzędzia przemysłowego	K_U08, Inż_U02, Inż_U04, Inż_U05																
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • D. Flanagan - JavaScript: The Definitive Guide, 6th Edition • S. Stefanov - Javascript Patterns • E. Hahn - Express in Action • M. Fogus - Functional Javascript • A. Beaulieu - Learning SQL 																
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <ul style="list-style-type: none"> • napisanie i prezentacja programu komputerowego • prezentacje rozwiązania zadania • prezentacja projektu programistycznego • egzamin pisemny 																
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Wykład, pracownia programistyczna - listy zadań pokrywające poszczególne partie materiału. Student zdobywa minimalną liczbę punktów określoną w regulaminie zajęć.</p> <p>Egzamin pisemny sprawdzający wiedzę w poszczególnych partiach materiału.</p>																
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="228 1339 1157 1406"> <tr> <td>pracownia</td> <td>30 godz.</td> </tr> <tr> <td>wykład</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="228 1485 1157 1619"> <tr> <td>przygotowanie do egzaminu</td> <td>20 godz.</td> </tr> <tr> <td>przygotowanie do pracowni</td> <td>40 godz.</td> </tr> <tr> <td>studiowanie tematyki wykładów i literatury</td> <td>20 godz.</td> </tr> <tr> <td>udział w egzaminie</td> <td>2 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="228 1697 1157 1765"> <tr> <td>Łączna liczba godzin</td> <td>142 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>6</td> </tr> </table>	pracownia	30 godz.	wykład	30 godz.	przygotowanie do egzaminu	20 godz.	przygotowanie do pracowni	40 godz.	studiowanie tematyki wykładów i literatury	20 godz.	udział w egzaminie	2 godz.	Łączna liczba godzin	142 godz.	Liczba punktów ECTS	6
pracownia	30 godz.																
wykład	30 godz.																
przygotowanie do egzaminu	20 godz.																
przygotowanie do pracowni	40 godz.																
studiowanie tematyki wykładów i literatury	20 godz.																
udział w egzaminie	2 godz.																
Łączna liczba godzin	142 godz.																
Liczba punktów ECTS	6																

4 Kursy narzędzi informatycznych (K1)

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs: Wstęp do programowania w języku C Introduction to Programming in C
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DK13WDPC
6.	Rodzaj przedmiotu/modułu Kurs podstawowy (K1)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Przydatna będzie podstawowa umiejętność programowania.
13.	Cele kształcenia dla przedmiotu Celem tego kursu jest nauczenie i rozwinięcie podstawowych umiejętności programowania w jednym z najbardziej rozpowszechnionych języków imperatywnych, tzn. w języku C. Wykład jest ilustrowany wieloma przykładami, a towarzyszące mu zajęcia w pracowniach uczą (poprzez wiele drobnych zadań i końcowy projekt) praktyki dobrego programowania.
14.	Treści programowe <ul style="list-style-type: none">• Podstawowe konstrukcje języka: instrukcje i deklaracje.• Standardowe typy danych, wyrażenia.• Standardowe wejście/wyjście, filtry.• Funkcje, moduły i struktura programu.• Metody agregacji danych: tablice, struktury i unie.• Wskaźniki.• Dynamiczne zarządzanie pamięcią, standardowe biblioteki C.• Sekwencyjne i swobodne przetwarzanie plików.• Programowanie okienkowego interfejsu w GTK+.• Klasy, obiekty i strumienie w C++.

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <hr/> <p>Zna podstawowe konstrukcje programistyczne oraz standardowe typy danych w języku C (przypisanie, instrukcje sterujące, wywoływanie funkcji i przekazywanie parametrów, typy całkowite i zmiennopozycyjne). Wie jak napisać proste sparametryzowane funkcje w C. Rozumie pojęcie składni języka programowania. Zna podstawowe struktury danych i wykonywane na nich operacje (tablice, napisy, rekordy, pliki, wskaźniki, struktury wskaźnikowe, listy, stosy, kolejki i drzewa).</p> <hr/> <p>Umiejętności</p> <hr/> <p>Potrafi pisać, uruchamiać i testować programy w C używając wybranego środowiska programistycznego. Potrafi wybrać i użyć w swoich programach podstawowe struktury danych. Umie podzielić program na moduły. Potrafi pisać funkcje rekurencyjne. Umie czytać ze zrozumieniem programy zapisane w języku C.</p>	<p>K_W05</p> <p>K_W05 K_W06</p> <p>K_U03</p> <p>K_U03 K_U03, K_U08 K_U03 K_U03, K_U09</p>
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • Kernighan B.W., Ritchie D.M.: Język ANSI C. Seria "Klasyka Informatyki". Warszawa, Wydawnictwa Naukowo-Techniczne 2003. Podręcznik podstawowy. • King K.N.: Język C, Nowoczesne programowanie, Wydanie II, Helion 2011 (uwzględnia standard C99). • Banahan M., Brady D., Doran M., The C Book, http://publications.gbdirect.co.uk/c_book/ (dostępny za darmo jako pdf i na WWW). • Schildt B., C: The Complete Reference: covers C++ and ANSI C, Osborne McGraw-Hill, • ISO/IEC 2011 - Programming languages - C (aktualny standard ISO 2011), http://www.open-std.org/JTC1/SC22/wg14/www/docs/n1570.pdf (wstępna wersja opisu standardu). 	
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>napisanie i prezentacja programu komputerowego; przygotowanie, zaprogramowanie i prezentacja projektu; kolokwium pisemne</p>	
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Zaliczenie wykładu:</p> <ul style="list-style-type: none"> • Kolokwium sprawdzające umiejętność programowania w C poprzez rozwiązanie 3 krótkich zadań programistycznych, w tym umiejętność napisania kompletnego programu i funkcji. Punkty uzyskane z kolokwium są doliczane do punktów z pracowni. <p>Zaliczenie pracowni:</p> <ul style="list-style-type: none"> • Listy zadań do samodzielnego programowania, wprowadzające kolejne elementy języka i oceniane przez prowadzących pracownię lub automatycznie przez system sprawdzający. • Końcowy projekt sprawdzający umiejętność zaprojektowania i zaprogramowania większego programu. Uzyskane punkty są sumowane z punktami z kolokwium i wystawiana jest jedna ocena końcowa. 	

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	30 godz.
	pracownia	30 godz.
	Praca własna studenta	
	udział w kolokwium	2 godz.
	przygotowanie do kolokwium	3 godz.
	przygotowanie do pracowni	30 godz.
	praca nad projektem	20 godz.
	studiowanie tematyki wykładów i literatury	10 godz.
Sumarycznie		
Łączna liczba godzin	125 godz.	
Liczba punktów ECTS	5	

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs programowania pod Windows w technologii .NET Design and development of .NET Windows Applications														
2.	Dyscyplina informatyka														
3.	Język wykładowy polski														
4.	Jednostka prowadząca przedmiot Instytut Informatyki														
5.	Kod przedmiotu/modułu 28-INF-S-DK6WNT														
6.	Rodzaj przedmiotu/modułu Kurs podstawowy (K1)														
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka														
8.	Poziom studiów studia I stopnia														
9.	Rok studiów ((jeśli obowiązuje)) —														
10.	Semestr letni														
11.	Forma zajęć i liczba godzin wykład – 30 godzin, pracownia – 30 godzin														
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu														
13.	Cele przedmiotu Celem przedmiotu jest zapoznanie słuchaczy z praktyką wytwarzania oprogramowania dla rodziny systemów Windows ze szczególnym uwzględnieniem technologii .NET.														
14.	Treści programowe <ul style="list-style-type: none"> • język C - klasy, właściwości, zdarzenia, delegaty, mechanizm typów generycznych, kontynuacje (yield), język LINQ, programowanie dynamiczne oraz rozszerzenia dla programowania asynchronicznego (async/await) • biblioteka standardowa platformy .NET - w szczególności technologie Windows.Forms, WPF, ADO.NET i XML • architektura systemu Windows, okna uchwyt, komunikaty • interfejs Win32 • technologia COM 														
15.	Zakładane efekty uczenia się <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2">Wiedza</td> </tr> <tr> <td style="border: none;">zna architekturę systemów rodziny Windows</td> <td style="border: none;">K_W07</td> </tr> <tr> <td style="border: none;">zna język C#</td> <td style="border: none;">K_W05</td> </tr> <tr> <td style="border: none;">zna przeglądowo bibliotekę standardową platformy .NET</td> <td style="border: none;">K_W07</td> </tr> <tr> <td colspan="2">Umiejętności</td> </tr> <tr> <td style="border: none;">potrafi wytworzyć aplikację przy użyciu platformy .NET, obejmującą interfejs użytkownika oraz komunikację z wybranym źródłem danych</td> <td style="border: none;">K_U03, K_U08</td> </tr> <tr> <td style="border: none;">potrafi napisać aplikację przy użyciu interfejsu Win32 w języku C/C++</td> <td style="border: none;">K_U03</td> </tr> </table>	Wiedza		zna architekturę systemów rodziny Windows	K_W07	zna język C#	K_W05	zna przeglądowo bibliotekę standardową platformy .NET	K_W07	Umiejętności		potrafi wytworzyć aplikację przy użyciu platformy .NET, obejmującą interfejs użytkownika oraz komunikację z wybranym źródłem danych	K_U03, K_U08	potrafi napisać aplikację przy użyciu interfejsu Win32 w języku C/C++	K_U03
Wiedza															
zna architekturę systemów rodziny Windows	K_W07														
zna język C#	K_W05														
zna przeglądowo bibliotekę standardową platformy .NET	K_W07														
Umiejętności															
potrafi wytworzyć aplikację przy użyciu platformy .NET, obejmującą interfejs użytkownika oraz komunikację z wybranym źródłem danych	K_U03, K_U08														
potrafi napisać aplikację przy użyciu interfejsu Win32 w języku C/C++	K_U03														

16.	Literatura obowiązkowa i zalecana	
	<ol style="list-style-type: none"> 1. Charles Petzold, Programming Windows 5th Edition 2. Mark Russinovich, David Solomon, Windows Internals 3. Bruce Eckel, Thinking in C# 4. Andrew Troelsen, Język C# i Platforma .NET 5. Daniel Solis, Illustrated C# 	
17.	Metody weryfikacji zakładanych efektów kształcenia	
	<ul style="list-style-type: none"> • napisanie i prezentacja programu komputerowego • prezentacje rozwiązania zadania 	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Wykład, pracownia programistyczna - listy zadań pokrywające poszczególne partie materiału. Student zdobywa minimalną liczbę punktów określoną w regulaminie zajęć.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
	przygotowanie do pracowni	40 godz.
	studiowanie tematyki wykładów i literatury	20 godz.
	Sumarycznie	
	Łączna liczba godzin	120 godz.
	Liczba punktów ECTS	5

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Introduction to cloud computing Introduction to cloud computing
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-ItCC
6.	Rodzaj przedmiotu/modułu Kurs podstawowy (K1)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin pracownia — 15 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu <ul style="list-style-type: none"> • General technical culture to learn technology independently. • Knowledge of the English language to allow. <ul style="list-style-type: none"> – Work with documentation and teaching materials, – presentation of own project, – group discussion.
13.	Cele kształcenia dla przedmiotu The primary goal of the course is to familiarize students with the cloud services currently offered on the market and to teach how to use them (at a basic level).
14.	Treści programowe <ol style="list-style-type: none"> 1. Review of existing cloud technologies and services offered by their providers. 2. Familiarization with selected cloud technologies and services offered in the cloud. 3. Designing and implementation of a cloud-based technology solution project (e.g. application, CD/CI solution, data science project) using selected cloud technologies and services offered in the cloud.

15.	Zakładane efekty uczenia się	
	Knowledge	
	knows the concept of cloud computing	K_W07
	knows the differences between the basic ways of using of the cloud - infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS) and software-as-a-service (SaaS)	K_W07
	knows the basic services offered in the cloud	K_W07
	knows where to look for knowledge about cloud services	K_W07
	Skills	
	is able to design and implement a simple technology solution (e.g., application, CD/CI solution, data science project) using selected cloud technologies and services offered in the cloud	K_U03, K_U08
	is able to independently learn cloud technologies	K_U13
	is able to present an idea of own project in an understandable and attractive way for the audience, and after its completion the project itself	K_U12
	is able to adequately address comments and criticism on their proposed solutions	K_U14
	is able to constructively criticize another students' projects	K_U14
	is able to speak English at the B2 level	K_U11
16.	Literatura obowiązkowa i zalecana	
	Teaching materials offered by cloud service providers	
	<ul style="list-style-type: none"> • Google Cloud Skills Boost https://www.cloudskillsboost.google/ • AWS Academy: Academy Cloud Foundations, Academy Learners Lab • inne materiały dostępne w internecie: aws.amazon.com, cloud.google.com 	
17.	Metody weryfikacji zakładanych efektów uczenia się	
	presentation of the project idea, presentation of the project, consultation on the progress of the project preparation, participation in the discussion on one's own project and the projects of other students	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Design and implementation of a cloud-based technology solution project (e.g., application, CD/CI solution, data science project) using cloud technologies and services offered in the cloud.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	presentation of own project and discussion on other students' projects	10 h
	presentation of own project idea and discussion on other students' project ideas	4 h
	Praca własna studenta	
	preparation of project proposal and its presentation	2 h
	participation in introductory classes	1 h
	study of selected technologies using the provided teaching materials	10 h
	preparation of the project	16 h
	preparation of the project presentation	2 h
	additional consultations as needed	2 h
	Sumarycznie	
	Łączna liczba godzin	47 godz.
	Liczba punktów ECTS	2

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs: Wstęp do programowania w języku Python Introduction to Programming in Python
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DK13WDPP
6.	Rodzaj przedmiotu/modułu Kurs podstawowy (K1)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Przedmiot nie ma żadnych wymagań wstępnych (adresowany jest do studentów pierwszego semestru)
13.	Cele kształcenia dla przedmiotu Przedmiot ma dwa główne cele. Pierwszym jest nauczenie studentów programowania w języku imperatywnym. Drugim celem jest przedstawienie specyficznych cech Pythona. Zakładamy, że dla części uczestników będzie to pierwszy język programowania (aczkolwiek wykład może być pożyteczny również dla osób umiejących programować i nie znających Pythona)

14.	<p>Treści programowe</p> <ol style="list-style-type: none"> 1. Podstawowe składniki programów: stałe, operatory, wyrażenia, zmienne, instrukcja podstawienia, sekwencje instrukcji, instrukcja warunkowa, pętla while, pętla for i in range(N) 2. Typy proste: int, float, typ logiczny, napisy, podstawowe operacje na wartościach tych typów, leniwe wartościowanie wyrażeń logicznych 3. Definiowanie funkcji, przekazywanie parametrów, możliwe semantyki instrukcji przypisania (przepisywanie referencji i kopiowanie), zmienne lokalne i globalne 4. Listy w Pythonie: podstawowe operacje, wycinki, iteracja po listach, pisanie funkcji (niezmieniających wartości argumentów) oraz procedur (funkcji nie zwracających wartości i mających efekty uboczne), różnica w pracy z obiektami zmiennymi i niezmiennymi na przykładzie list i napisów, tablice dwuwymiarowe implementowane jako listy list, implementacja prostych algorytmów sortowania 5. Biblioteka turtle jako przykład prostej biblioteki umożliwiającej tworzenie rysunków, model RGB kolorów 6. Zbiory oraz operacje na zbiorach, wykorzystanie bibliotecznego sortowania, porównanie efektywności czasowej różnych rozwiązań, ze szczególnym naciskiem na kwadratową złożoność pojedynczej pętli i na sposoby unikania tego problemu 7. elementy programowania funkcyjnego: listy składane, map, filter, funkcje jako wartości pierwszego rzędu, wyrażenia funkcyjne (lambda), rekurencja w definiowaniu funkcji, funkcje all i any, funkcje z różną liczbą argumentów 8. Podstawowe operacje na słownikach, praktyczne programy korzystające z dużych słowników (prosty program tłumaczący z polskiego na angielski), tworzenie słowników z wartościami domyślnymi (defaultdict z modułu collections), słownik jako rzadka reprezentacja macierzy (krotki jako klucze) 9. Czytanie i pisanie danych do plików, iteracja po pliku 10. Programy jako dane: wykorzystywanie funkcji eval i exec, wyjątki i ich przechwytywanie, tworzenie prostych interpreterów różnych języków bazujących na ww. funkcjach i przekształcaniu napisów 11. Drzewa i rekurencyjne struktury danych, grafy jako słowniki, proste algorytmy grafowe (przeszukiwanie włąb, wszere, wyszukiwanie składowych) 12. yield i tworzenie funkcji iteratorów, elementy programowania obiektowego (obiekty, konstruktory, klasy, przeciążanie operatorów w celu tworzenia wariantów typów standardowych (liczby wymierne, listy z dowiązaniem do interfejsu analogicznym do standardowych list) 																								
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 1261 1428 1485"> <tr> <td>zna podstawowe proste i złożone typy danych używane w językach programowania</td> <td>K_W05</td> </tr> <tr> <td>zna podstawowe instrukcje tworzące programy imperatywne</td> <td>K_W05</td> </tr> <tr> <td>zna podstawy języka Python, podstawy programowania obiektowego i funkcyjnego</td> <td>K_W05</td> </tr> <tr> <td>zna praktyczne aspekty złożoności pamięciowej i czasowej programów pisanych w języku wysokiego poziomu</td> <td>K_W05</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 1570 1428 1794"> <tr> <td>umie samodzielnie rozwiązywać z języku Python różne zadania programistyczne</td> <td>K_U03</td> </tr> <tr> <td>umie analizować i modyfikować kod w języku Python napisany przez inną osobę</td> <td>K_U03</td> </tr> <tr> <td>potrafi łączyć imperatywny styl programowania z elementami programowania funkcyjnego</td> <td>K_U03</td> </tr> <tr> <td>umie pisać programy tworzące obrazy, przetwarzające tekst, dane liczbowe, operujące na plikach</td> <td>K_U03, K_U08</td> </tr> <tr> <td>umie pisać programy wykorzystujące zaawansowane struktury danych</td> <td>K_U03, K_U08</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 1879 1428 2074"> <tr> <td>student rozumie znaczenie samodzielnej pracy przy zdobywaniu nowych umiejętności</td> <td>K_K05</td> </tr> <tr> <td>student umie zadawać pytania przy napotkaniu trudności przy rozwiązywaniu jakiegoś zadania</td> <td>K_K06</td> </tr> <tr> <td>student umie opowiedzieć o swoim rozwiązaniu jakiegoś problemu skupiając się na najbardziej istotnych dla słuchacza elementach</td> <td>K_K04</td> </tr> </table>	zna podstawowe proste i złożone typy danych używane w językach programowania	K_W05	zna podstawowe instrukcje tworzące programy imperatywne	K_W05	zna podstawy języka Python, podstawy programowania obiektowego i funkcyjnego	K_W05	zna praktyczne aspekty złożoności pamięciowej i czasowej programów pisanych w języku wysokiego poziomu	K_W05	umie samodzielnie rozwiązywać z języku Python różne zadania programistyczne	K_U03	umie analizować i modyfikować kod w języku Python napisany przez inną osobę	K_U03	potrafi łączyć imperatywny styl programowania z elementami programowania funkcyjnego	K_U03	umie pisać programy tworzące obrazy, przetwarzające tekst, dane liczbowe, operujące na plikach	K_U03, K_U08	umie pisać programy wykorzystujące zaawansowane struktury danych	K_U03, K_U08	student rozumie znaczenie samodzielnej pracy przy zdobywaniu nowych umiejętności	K_K05	student umie zadawać pytania przy napotkaniu trudności przy rozwiązywaniu jakiegoś zadania	K_K06	student umie opowiedzieć o swoim rozwiązaniu jakiegoś problemu skupiając się na najbardziej istotnych dla słuchacza elementach	K_K04
zna podstawowe proste i złożone typy danych używane w językach programowania	K_W05																								
zna podstawowe instrukcje tworzące programy imperatywne	K_W05																								
zna podstawy języka Python, podstawy programowania obiektowego i funkcyjnego	K_W05																								
zna praktyczne aspekty złożoności pamięciowej i czasowej programów pisanych w języku wysokiego poziomu	K_W05																								
umie samodzielnie rozwiązywać z języku Python różne zadania programistyczne	K_U03																								
umie analizować i modyfikować kod w języku Python napisany przez inną osobę	K_U03																								
potrafi łączyć imperatywny styl programowania z elementami programowania funkcyjnego	K_U03																								
umie pisać programy tworzące obrazy, przetwarzające tekst, dane liczbowe, operujące na plikach	K_U03, K_U08																								
umie pisać programy wykorzystujące zaawansowane struktury danych	K_U03, K_U08																								
student rozumie znaczenie samodzielnej pracy przy zdobywaniu nowych umiejętności	K_K05																								
student umie zadawać pytania przy napotkaniu trudności przy rozwiązywaniu jakiegoś zadania	K_K06																								
student umie opowiedzieć o swoim rozwiązaniu jakiegoś problemu skupiając się na najbardziej istotnych dla słuchacza elementach	K_K04																								

16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • Materiały na stronie wykładu (wszystkie zadania da się rozwiązać, korzystając z tego źródła wiedzy). <p>Literatura uzupełniająca</p> <ul style="list-style-type: none"> • Materiały ze strony python.org • Dowolny podręcznik Pythona dla początkujących, na przykład: Michael Dawson, „Python dla każdego. Podstawy programowania.” 																
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>Prezentacja programu napisanego w domu, samodzielne pisanie programów na zajęciach (tzw. wprawki programistyczne), kolokwium</p>																
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <ul style="list-style-type: none"> • należy zdobyć wymaganą, określoną w regulaminie przedmiotu łączną liczbę punktów za zadania domowe, wprawki programistyczne (czyli programy pisane na zajęciach) oraz kolokwium • dodatkowo konieczne jest zdobycie określonej w regulaminie minimalnej liczby punktów za kolokwium 																
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 790 1428 857"> <tr> <td>wykład</td> <td>30 godz.</td> </tr> <tr> <td>pracownia</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 936 1428 1059"> <tr> <td>Przygotowanie do kolokwium</td> <td>15 godz.</td> </tr> <tr> <td>Studiowanie materiałów dodatkowych</td> <td>15 godz.</td> </tr> <tr> <td>Samodzielna analiza programów z wykładu</td> <td>10 godz.</td> </tr> <tr> <td>Dodatkowa praca nad zadaniami w domu</td> <td>30 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 1149 1428 1216"> <tr> <td>Łączna liczba godzin</td> <td>130 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>5</td> </tr> </table>	wykład	30 godz.	pracownia	30 godz.	Przygotowanie do kolokwium	15 godz.	Studiowanie materiałów dodatkowych	15 godz.	Samodzielna analiza programów z wykładu	10 godz.	Dodatkowa praca nad zadaniami w domu	30 godz.	Łączna liczba godzin	130 godz.	Liczba punktów ECTS	5
wykład	30 godz.																
pracownia	30 godz.																
Przygotowanie do kolokwium	15 godz.																
Studiowanie materiałów dodatkowych	15 godz.																
Samodzielna analiza programów z wykładu	10 godz.																
Dodatkowa praca nad zadaniami w domu	30 godz.																
Łączna liczba godzin	130 godz.																
Liczba punktów ECTS	5																

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs: Podstawowy warsztat informatyka Computing for Computer Scientists
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DKPWI
6.	Rodzaj przedmiotu/modułu Kurs podstawowy (K1)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 10 godzin, pracownia — 10 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Umiejętność obsługi komputera
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest przybliżenie studentom pierwszego roku podstawowych narzędzi do współtworzenia, kompilowania, uruchamiania, publikowania oraz dokumentowania swoich oraz cudzych programów.
14.	Treści programowe <ol style="list-style-type: none"> 1. Praca w systemie Linux: pliki, procesy, użytkownicy, potoki i strumienie, kompilowanie i uruchamianie programów, bash. 2. Polecenie ssh oraz podstawy kryptografii (klucze prywatne i publiczne) 3. Systemy kontroli wersji na przykładzie systemu git. 4. Podstawy pracy grupowej. 5. LaTeX (poprawne składanie tekstów takich jak rozwiązanie zadania, cv i list motywacyjny, wyrażenia matematyczne).

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <hr/> <p>zna i rozumie sposoby wykorzystywania narzędzi informatycznych wymaganych w pracy informatyka posiada wiedzę o podstawowych narzędziach kryptograficznych, zna i rozumie podstawowe zagadnienia związane z bezpieczeństwem komputerowym z perspektywy użytkownika rozumie zalety korzystania z systemów kontroli wersji: zna i rozumie filozofię systemu git</p> <hr/> <p>Umiejętności</p> <hr/> <p>potrafi używać systemu Linux jest w stanie samodzielnie nauczyć się wykorzystywania nieznanego mu narzędzia informatycznego na podstawie dokumentacji oraz samodzielnego wyszukiwania informacji w internecie posiada umiejętność krytycznego podejścia do informacji w wyszukanych przez siebie źródłach używa podstawowych narzędzi kryptograficznych potrafi używać narzędzi pozwalających na aktywny udział w projektach programistycznych (git i serwisy typu github) potrafi składać tekst z wykorzystaniem systemu LaTeX</p>	<p>K_W07</p> <p>K_W07</p> <p>K_W07</p> <hr/> <p>K_U10</p> <p>K_U13</p> <p>K_U13</p> <p>K_U07</p> <p>K_U07</p> <p>K_U07</p>
16.	<p>Literatura obowiązkowa i zalecana</p> <p>W związku z deklarowanym celem przedmiotu - nauczanie studentów umiejętności samodzielnego znajdowania odpowiednich źródeł informacji, a także krytycznego podejścia do tych źródeł - nie jest podana literatura obowiązkowa.</p> <p>Zalecane źródła: dokumentacja narzędzi informatycznych, podręczniki użytkownika, specjalistyczne fora dyskusyjne itp.</p> <ul style="list-style-type: none"> • Official Ubuntu Documentation https://help.ubuntu.com/ • GNU Bash Manual https://www.gnu.org/software/bash/manual/ • git User Manual https://git-scm.com/doc • OpenSSH Manual https://www.openssh.com/manual.html • LaTeX Documentation https://www.latex-project.org/help/documentation/ • The beamer class User Guide https://github.com/josephwright/beamer 	
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <ul style="list-style-type: none"> • prezentacja rozwiązania zadania • test 	
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Do zaliczenia pracowni należy zdobyć wymaganą, podaną w regulaminie przedmiotu liczbę punktów za rozwiązanie zadań na pracownię oraz za napisanie testu.</p>	
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <hr/> <p>wykład pracownia</p> <hr/> <p>Praca własna studenta</p> <hr/> <p>przygotowywanie się do pracowni (rozwiązywanie zadań, wyszukiwanie informacji w dokumentacji oraz w innych źródłach, czytanie materiałów uzupełniających) przygotowanie projektu</p> <hr/> <p>Sumarycznie</p> <hr/> <p>Łączna liczba godzin Liczba punktów ECTS</p>	<p>10 godz. 10 godz.</p> <hr/> <p>30 godz. 25 godz.</p> <hr/> <p>75 godz. 3</p>

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs języka C++ C++ course
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DK6CPP
6.	Rodzaj przedmiotu/modułu Kurs podstawowy (K1)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: <ul style="list-style-type: none"> • kurs języka ANSI C z elementami C++ Niezbędne kompetencje: <ul style="list-style-type: none"> • umiejętność programowania strukturalnego i proceduralnego w języku ANSI C; • znajomość podstawowych struktur danych; • umiejętność czytania anglojęzycznej dokumentacji.
13.	Cele kształcenia dla przedmiotu Celem tego kursu jest nauczanie programowania obiektowego w bardzo popularnym współcześnie języku programowania C++. Na kursie jest prezentowany najnowszy standard tego języka oraz obszerne fragmenty biblioteki standardowej STL. Wykład jest ilustrowany programami napisanymi zgodnie z najlepszymi wzorcami, natomiast towarzyszące zajęcia laboratoryjne mają nauczyć dobrych praktyk w programowaniu i projektowaniu obiekto- wym.

14.	<p>Treści programowe</p> <ol style="list-style-type: none"> 1. Podstawowe konstrukcje językowe w C++. 2. Abstrakcja i hermetyzacja. 3. Inicjalizacja, kopiowanie, przenoszenie. 4. Składowe statyczne. 5. Przeciążanie operatorów. 6. Dziedziczenie, wielodziedziczenie. 7. Polimorfizm, klasy abstrakcyjne. 8. Przestrzenie nazw. 9. Wyjątki, asercje. 10. Szablony funkcji i szablony klas. 11. Konwersje. 12. Strumienie, operacje na plikach. 13. Podstawowe kolekcje standardowe. 14. Podstawowe algorytmy. 15. Operacje na łańcuchach znakowych. 																						
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">zna standardowe typy danych w języku C++ oraz podstawowe konstrukcje programistyczne (instrukcje sterujące, funkcje) wykorzystywane w programowaniu imperatywnym</td> <td style="width: 20%;">K_W05</td> </tr> <tr> <td>zna proste i złożone typy danych używane w języku C++</td> <td>K_W05</td> </tr> <tr> <td>zna zasady programowania obiektowego w C++</td> <td>K_W05</td> </tr> <tr> <td>zna praktyczne aspekty złożoności pamięciowej i czasowej programów pisanych w języku wysokiego poziomu</td> <td>K_W05</td> </tr> </table> <p>Umiejętności</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">potrafi pisać, uruchamiać i testować programy napisane w C++, używając wybranego środowiska programistycznego</td> <td style="width: 20%;">K_U03</td> </tr> <tr> <td>umie samodzielnie rozwiązywać różne zadania programistyczne korzystając z języka C++</td> <td>K_U03</td> </tr> <tr> <td>potrafi wybrać i użyć w swoich programach podstawowych i złożonych struktur danych</td> <td>K_U03, K_U09</td> </tr> <tr> <td>umie podzielić program w C++ na moduły</td> <td>K_U03, K_U08</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Student rozumie znaczenie samodzielnej pracy przy zdobywaniu nowych umiejętności</td> <td style="width: 20%;">K_K05</td> </tr> <tr> <td>Student umie zadawać pytania dotyczące rozwiązywania jakiegoś zadania</td> <td>K_K06</td> </tr> <tr> <td>Student umie opowiedzieć o swoim rozwiązaniu koncentrując się na najbardziej istotnych zagadnieniach</td> <td>K_K04</td> </tr> </table>	zna standardowe typy danych w języku C++ oraz podstawowe konstrukcje programistyczne (instrukcje sterujące, funkcje) wykorzystywane w programowaniu imperatywnym	K_W05	zna proste i złożone typy danych używane w języku C++	K_W05	zna zasady programowania obiektowego w C++	K_W05	zna praktyczne aspekty złożoności pamięciowej i czasowej programów pisanych w języku wysokiego poziomu	K_W05	potrafi pisać, uruchamiać i testować programy napisane w C++, używając wybranego środowiska programistycznego	K_U03	umie samodzielnie rozwiązywać różne zadania programistyczne korzystając z języka C++	K_U03	potrafi wybrać i użyć w swoich programach podstawowych i złożonych struktur danych	K_U03, K_U09	umie podzielić program w C++ na moduły	K_U03, K_U08	Student rozumie znaczenie samodzielnej pracy przy zdobywaniu nowych umiejętności	K_K05	Student umie zadawać pytania dotyczące rozwiązywania jakiegoś zadania	K_K06	Student umie opowiedzieć o swoim rozwiązaniu koncentrując się na najbardziej istotnych zagadnieniach	K_K04
zna standardowe typy danych w języku C++ oraz podstawowe konstrukcje programistyczne (instrukcje sterujące, funkcje) wykorzystywane w programowaniu imperatywnym	K_W05																						
zna proste i złożone typy danych używane w języku C++	K_W05																						
zna zasady programowania obiektowego w C++	K_W05																						
zna praktyczne aspekty złożoności pamięciowej i czasowej programów pisanych w języku wysokiego poziomu	K_W05																						
potrafi pisać, uruchamiać i testować programy napisane w C++, używając wybranego środowiska programistycznego	K_U03																						
umie samodzielnie rozwiązywać różne zadania programistyczne korzystając z języka C++	K_U03																						
potrafi wybrać i użyć w swoich programach podstawowych i złożonych struktur danych	K_U03, K_U09																						
umie podzielić program w C++ na moduły	K_U03, K_U08																						
Student rozumie znaczenie samodzielnej pracy przy zdobywaniu nowych umiejętności	K_K05																						
Student umie zadawać pytania dotyczące rozwiązywania jakiegoś zadania	K_K06																						
Student umie opowiedzieć o swoim rozwiązaniu koncentrując się na najbardziej istotnych zagadnieniach	K_K04																						
16.	<p>Literatura obowiązkowa i zalecana</p> <p>Literatura podstawowa:</p> <ul style="list-style-type: none"> • B.Stroustrup: Język C++. Kompendium wiedzy. Wydanie 4. Wydawnictwo Helion, Gliwice 2014. • J.Grębosz: Opus magnum C++11. Programowanie w języku C++. Tom 1, 2, 3. Wydawnictwo Helion, Gliwice 2018. <p>Literatura uzupełniająca:</p> <ul style="list-style-type: none"> • S.Rao: C++. Dla każdego. Wydanie 7. Wydawnictwo Helion, Gliwice 2014. • S.Prata: Język C++. Szkoła programowania. Wydanie 6. Wydawnictwo Helion, Gliwice 2012. 																						
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>Pisanie i prezentacja krótkich programów komputerowych i małych aplikacji (miniprojekty).</p>																						
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Do zaliczenia przedmiotu należy zdobyć minimalną wymaganą (podaną w regulaminie przedmiotu) liczbę punktów, które uzyskuje się za wykonane w ramach pracowni programy.</p>																						

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	30 godz.
	pracownia	30 godz.
	Praca własna studenta	
	przygotowanie do pracowni	45 godz.
	czytanie literatury i dokumentacji	15 godz.
	Sumarycznie	
	Łączna liczba godzin	120 godz.
	Liczba punktów ECTS	5

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs rozszerzony języka Python Advanced Python Programming
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DK11RKJP
6.	Rodzaj przedmiotu/modułu Kurs podstawowy (K1)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu <ul style="list-style-type: none"> • Znajomość podstawowych struktur danych: listy, kolejki, drzewa binarne. • Podstawowa umiejętność programowania w Pythonie.
13.	Cele kształcenia dla przedmiotu Celem zajęć jest <ul style="list-style-type: none"> • rozwijanie wiedzy i umiejętności korzystania z zaawansowanych cech języka (listy składane, generatory, metaprogramowanie) • nabycie umiejętności posługiwania się standardowymi bibliotekami: programowanie wielowątkowe, usługi sieciowe, interfejs użytkownika, dostęp i przechowywanie danych • poznanie narzędzi wspierających rozwój projektów (testowanie jednostkowe, profilowanie, metryki) • przegląd popularnych zastosowań Pythona wraz ze stosowanymi narzędziami: aplikacje webowe, przetwarzanie danych, uczenie maszynowe
14.	Treści programowe <ul style="list-style-type: none"> • listy składane, generatory • obiekty w nowym stylu, metaklasy • wątki, procesy • interfejsy użytkownika • testowanie programów, standard PEP 8 • programowanie sieciowe • trwałe słowniki, Python DB API • django, matplotlib, pandas, PyTorch/TensorFlow

15.	Zakładane efekty uczenia się												
	Wiedza												
	<table border="1"> <tr> <td>zna składnię i semantykę języka Python</td> <td>K_W05</td> </tr> <tr> <td>zna najpopularniejsze moduły dostępne w Pythonie</td> <td>K_W07</td> </tr> </table>	zna składnię i semantykę języka Python	K_W05	zna najpopularniejsze moduły dostępne w Pythonie	K_W07								
zna składnię i semantykę języka Python	K_W05												
zna najpopularniejsze moduły dostępne w Pythonie	K_W07												
	Umiejętności												
	<table border="1"> <tr> <td>potrafi implementować zadania wykorzystując zaawansowane właściwości Pythona, takie jak wbudowane struktury danych, listy składane czy obiekty w nowym stylu</td> <td>K_U03, K_U08</td> </tr> <tr> <td>potrafi implementować w Pythonie różne funkcjonalności aplikacji (trwale przechowywanie danych, implementacja interfejsu użytkownika)</td> <td>K_U03, K_U08</td> </tr> <tr> <td>potrafi zadbać o jakość kodu i zna rekomendacje dot. nazewnictwa i formatowania kodu</td> <td>K_U03, K_U09</td> </tr> <tr> <td>potrafi zaprogramować aplikację internetową z wykorzystaniem odpowiedniej platformy (np. django)</td> <td>K_U03, K_U08</td> </tr> <tr> <td>umie opracować i zaprezentować architekturę rozwiązania informatycznego</td> <td>K_U10</td> </tr> </table>	potrafi implementować zadania wykorzystując zaawansowane właściwości Pythona, takie jak wbudowane struktury danych, listy składane czy obiekty w nowym stylu	K_U03, K_U08	potrafi implementować w Pythonie różne funkcjonalności aplikacji (trwale przechowywanie danych, implementacja interfejsu użytkownika)	K_U03, K_U08	potrafi zadbać o jakość kodu i zna rekomendacje dot. nazewnictwa i formatowania kodu	K_U03, K_U09	potrafi zaprogramować aplikację internetową z wykorzystaniem odpowiedniej platformy (np. django)	K_U03, K_U08	umie opracować i zaprezentować architekturę rozwiązania informatycznego	K_U10		
potrafi implementować zadania wykorzystując zaawansowane właściwości Pythona, takie jak wbudowane struktury danych, listy składane czy obiekty w nowym stylu	K_U03, K_U08												
potrafi implementować w Pythonie różne funkcjonalności aplikacji (trwale przechowywanie danych, implementacja interfejsu użytkownika)	K_U03, K_U08												
potrafi zadbać o jakość kodu i zna rekomendacje dot. nazewnictwa i formatowania kodu	K_U03, K_U09												
potrafi zaprogramować aplikację internetową z wykorzystaniem odpowiedniej platformy (np. django)	K_U03, K_U08												
umie opracować i zaprezentować architekturę rozwiązania informatycznego	K_U10												
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • Tarek Ziade - Expert Python Programming • Dusty Phillips - Python 3 Object Oriented Programming • Rick Copeland- Essential SQLAlchemy • A. Holovaty, J. Kaplan-Moss - The Django Book. Release 2.0 • An introduction to Python programming with NumPy, SciPy and Matplotlib/PyLab - Antoine Lefebvre 												
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <ul style="list-style-type: none"> • implementacja zadań • realizacja bardziej złożonego projektu 												
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Do zaliczenia pracowni należy zdobyć wskazaną w regulaminie zajęć liczbę punktów za implementację programów oraz za zrealizowanie projektu.</p>												
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1"> <tr> <td>wykład</td> <td>30 godz.</td> </tr> <tr> <td>pracownia</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1"> <tr> <td>Implementacja zadań</td> <td>45 godz.</td> </tr> <tr> <td>realizacja i prezentacja projektu</td> <td>15 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1"> <tr> <td>Łączna liczba godzin</td> <td>120 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>5</td> </tr> </table>	wykład	30 godz.	pracownia	30 godz.	Implementacja zadań	45 godz.	realizacja i prezentacja projektu	15 godz.	Łączna liczba godzin	120 godz.	Liczba punktów ECTS	5
wykład	30 godz.												
pracownia	30 godz.												
Implementacja zadań	45 godz.												
realizacja i prezentacja projektu	15 godz.												
Łączna liczba godzin	120 godz.												
Liczba punktów ECTS	5												

5 Kursy narzędzi informatycznych inżynierskie (Kinż)

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs projektowania aplikacji z bazami danych Design of database applications
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DK15PABD
6.	Rodzaj przedmiotu/modułu Kurs inżynierski (Kinż)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów ((jeśli obowiązuje)) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 30 godzin, pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Umiejętność programowania na dowolnej platformie programistycznej oraz znajomość materiału z kursu WWW.
13.	Cele przedmiotu Dobrze zaprojektowana aplikacja przetwarzająca i prezentująca dane ma wiele warstw i komponentów o określonych odpowiedzialnościach. Głównym celem zajęć jest zaprezentowanie, jak dobrze taki typ aplikacji zaprojektować i oprogramować. Drugim celem jest wprowadzenie w świat baz danych na przykładzie SQL Servera.

14.	<p>Treści programowe</p> <p>Część 1: DBMS</p> <ol style="list-style-type: none"> 1. Podstawy Microsoft SQL Server 2. Język SQL 3. Programowanie w T-SQL w tym kursory 4. Wyzwalacze, funkcje i procedury 5. Transakcje i blokady 6. Optymalizacja w tym normalizacja, indeksy i plany wykonania 7. Inne rodzaje baz danych, w tym grafowe i tekstowe <p>Część 2: System informatyczny</p> <ol style="list-style-type: none"> 1. Podstawy uruchomienia rozwiązania w chmurze 2. Architektura Microservices, podstawy konteneryzacji 3. Architektury aplikacji z bazą danych 4. Przegląd Domain-Driven Design 5. Podstawy testowania 6. Wzorzec repozytorium, w tym narzędzia ORM i LINQ 7. Walidacja danych 8. Modele danych i automapper 9. Prezentacja danych <ul style="list-style-type: none"> * sposoby prezentacji informacji * stronicowanie, sortowanie, filtry * wzorzec specyfikacji 10. Integracja systemów <ul style="list-style-type: none"> * wzorce integracyjne * usługi danych, protokół OData * API management 11. Podstawy federacji tożsamości 12. Wzorce CQRS i Event Sourcing 13. Skalowalność rozwiązań 14. Zarządzanie transakcjami, transakcje rozproszone 15. Podstawy analizy danych i raportowania 																						
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">Zna podstawowe pojęcia baz danych takie jak język SQL, programowanie w np. T-SQL, kursory, wyzwalacze, funkcje i procedury, transakcje i blokady, postacie normalne, indeksy oraz plany wykonania</td> <td style="padding: 5px;">K_W07, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">Zna podstawowe pojęcia nierelacyjnych baz danych</td> <td style="padding: 5px;">K_W08, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">Zna architekturę oraz podstawowe warstwy i komponenty systemu przetwarzającego dane</td> <td style="padding: 5px;">K_W07, K_K02, K_K04, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">Zna podstawy DDD i modelowania systemów</td> <td style="padding: 5px;">K_W07, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">Zna podstawowe wzorce i mechanizmy integracji systemów</td> <td style="padding: 5px;">K_W07, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">Zna zasady udostępniania i prezentacji danych</td> <td style="padding: 5px;">K_W07, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">Zna wzorce i podstawy skalowania systemów</td> <td style="padding: 5px;">K_W07, K_W08, Inż_W01</td> </tr> </table> <p>Umiejętności</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">Potrafi zaprojektować i zaimplementować relacyjną i nierelacyjną bazę danych</td> <td style="padding: 5px;">K_U03, K_U07, K_U08, K_U09, Inż_U01</td> </tr> <tr> <td style="padding: 5px;">Potrafi poprawnie przeprowadzić analizę wymagań dla prostego systemu przetwarzającego dane</td> <td style="padding: 5px;">K_U07, K_U09, Inż_U01, Inż_U03, Inż_U04, Inż_U05</td> </tr> <tr> <td style="padding: 5px;">Potrafi poprawnie zaprojektować prosty system przetwarzający dane</td> <td style="padding: 5px;">Inż_U03, K_U07, Inż_U01</td> </tr> <tr> <td style="padding: 5px;">Potrafi poprawnie zaimplementować, przetestować i wdrożyć prosty system przetwarzający dane</td> <td style="padding: 5px;">K_U03, K_U08</td> </tr> </table>	Zna podstawowe pojęcia baz danych takie jak język SQL, programowanie w np. T-SQL, kursory, wyzwalacze, funkcje i procedury, transakcje i blokady, postacie normalne, indeksy oraz plany wykonania	K_W07, Inż_W01	Zna podstawowe pojęcia nierelacyjnych baz danych	K_W08, Inż_W01	Zna architekturę oraz podstawowe warstwy i komponenty systemu przetwarzającego dane	K_W07, K_K02, K_K04, Inż_W01	Zna podstawy DDD i modelowania systemów	K_W07, Inż_W01	Zna podstawowe wzorce i mechanizmy integracji systemów	K_W07, Inż_W01	Zna zasady udostępniania i prezentacji danych	K_W07, Inż_W01	Zna wzorce i podstawy skalowania systemów	K_W07, K_W08, Inż_W01	Potrafi zaprojektować i zaimplementować relacyjną i nierelacyjną bazę danych	K_U03, K_U07, K_U08, K_U09, Inż_U01	Potrafi poprawnie przeprowadzić analizę wymagań dla prostego systemu przetwarzającego dane	K_U07, K_U09, Inż_U01, Inż_U03, Inż_U04, Inż_U05	Potrafi poprawnie zaprojektować prosty system przetwarzający dane	Inż_U03, K_U07, Inż_U01	Potrafi poprawnie zaimplementować, przetestować i wdrożyć prosty system przetwarzający dane	K_U03, K_U08
Zna podstawowe pojęcia baz danych takie jak język SQL, programowanie w np. T-SQL, kursory, wyzwalacze, funkcje i procedury, transakcje i blokady, postacie normalne, indeksy oraz plany wykonania	K_W07, Inż_W01																						
Zna podstawowe pojęcia nierelacyjnych baz danych	K_W08, Inż_W01																						
Zna architekturę oraz podstawowe warstwy i komponenty systemu przetwarzającego dane	K_W07, K_K02, K_K04, Inż_W01																						
Zna podstawy DDD i modelowania systemów	K_W07, Inż_W01																						
Zna podstawowe wzorce i mechanizmy integracji systemów	K_W07, Inż_W01																						
Zna zasady udostępniania i prezentacji danych	K_W07, Inż_W01																						
Zna wzorce i podstawy skalowania systemów	K_W07, K_W08, Inż_W01																						
Potrafi zaprojektować i zaimplementować relacyjną i nierelacyjną bazę danych	K_U03, K_U07, K_U08, K_U09, Inż_U01																						
Potrafi poprawnie przeprowadzić analizę wymagań dla prostego systemu przetwarzającego dane	K_U07, K_U09, Inż_U01, Inż_U03, Inż_U04, Inż_U05																						
Potrafi poprawnie zaprojektować prosty system przetwarzający dane	Inż_U03, K_U07, Inż_U01																						
Potrafi poprawnie zaimplementować, przetestować i wdrożyć prosty system przetwarzający dane	K_U03, K_U08																						

16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • Vernon Vaughn, Implementing Domain-Driven Design, Addison-Wesley Professional, 2013 • Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, Third Edition, The McGraw-Hill Companies, 2009 • Miguel Cebollero, Michael Coles, Jay Natarajan, Pro T-SQL Programmer's Guide, Apress, 2015 • Kathi Kellenberger, Scott Shaw, Beginning T-SQL, Apress, 2014 • Danuta Mendrala, Marcin Szeliga, Praktyczny kurs SQL. Wydanie III, Helion, 2015 												
17.	<p>Metody weryfikacji zakładanych efektów kształcenia</p> <p>Prezentacja rozwiązania zadań</p>												
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Pracownia: zaliczenie kolejnych list zadań, które prowadzą do kompletnego rozwiązania informatycznego.</p>												
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1"> <tr> <td>pracownia</td> <td>30 godz.</td> </tr> <tr> <td>wykład</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1"> <tr> <td>przygotowanie do pracowni</td> <td>55 godz.</td> </tr> <tr> <td>studiowanie materiału z wykładu</td> <td>20 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1"> <tr> <td>Łączna liczba godzin</td> <td>135 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>5</td> </tr> </table>	pracownia	30 godz.	wykład	30 godz.	przygotowanie do pracowni	55 godz.	studiowanie materiału z wykładu	20 godz.	Łączna liczba godzin	135 godz.	Liczba punktów ECTS	5
pracownia	30 godz.												
wykład	30 godz.												
przygotowanie do pracowni	55 godz.												
studiowanie materiału z wykładu	20 godz.												
Łączna liczba godzin	135 godz.												
Liczba punktów ECTS	5												

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs: Praktyczne aspekty rozwoju oprogramowania Practical aspects in software development
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DKPARO
6.	Rodzaj przedmiotu/modułu Kurs inżynierski (Kinż)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów ((jeśli obowiązuje)) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 15 godzin, pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Niezbędne kompetencje • Podstawowa znajomość C++
13.	Cele przedmiotu Nabycie wiedzy w zakresie rozwoju oprogramowania przy użyciu języka C++.
14.	Treści programowe 1. Jak realizować innowacyjne projekty na czas [Wykład] 2. Metodyki zwinne w zmiennym środowisku projektowym [Warsztat] 3. Środowisko programisty C++ [Pracownia] 4. Zarządzanie pamięcią w C++ [Pracownia] 5. Biblioteka STL w C++ [Pracownia] 6. Nowoczesne C++ (C++11/14/17) [Pracownia] 7. Test Driven Development na przykładzie C++ [Pracownia] 8. Tworzenie SOLIDnego kodu obiektowego w C++ [Pracownia] 9. Wzorce projektowe na przykładzie C++ [Pracownia] 10. Programowanie współbieżne w C++ [Pracownia] 11. Optymalizacje w C++ [Pracownia] 12. Coding Dojo w języku C++ [Pracownia] 13. Warsztat zwinnego programisty [Warsztat] 14. Kolokwium [Kolokwium] 15. Budowanie złożonych systemów informatycznych [Wykład]

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="228 192 1153 320"> <tr> <td>zna założenia metodyk zwinnych w odniesieniu do projektów programistycznych</td> <td>K_W07, K_W08, Inż_W01</td> </tr> <tr> <td>zna wybrane wzorce projektowe stosowane w rozwoju oprogramowania</td> <td>K_W07, Inż_W01</td> </tr> <tr> <td>zna podstawowe metody pracy z kodem obiektowo zorientowanym</td> <td>K_W07, Inż_W01</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="228 405 1153 622"> <tr> <td>potrafi posługiwać się zintegrowanym środowiskiem do rozwoju oprogramowania</td> <td>K_U03, K_U07, Inż_U03, Inż_U05</td> </tr> <tr> <td>potrafi napisać testy jednostkowe przy użyciu wybranej platformy testowej</td> <td>K_U08, K_U10, K_U13, Inż_U01, Inż_U02</td> </tr> <tr> <td>potrafi korzystać z wybranych elementów języka C++ w praktyce</td> <td>K_U03, K_U07, Inż_U04</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="228 707 1153 835"> <tr> <td>ma świadomość potrzeby dzielenia się wiedzą w ramach zespołu projektowego</td> <td>K_K06</td> </tr> <tr> <td>rozumie potrzebę dbania o jakość oprogramowania w kontekście pracy zespołowej</td> <td>K_K06</td> </tr> <tr> <td>rozumie konieczność stałego dokształcania się w obliczu ciągłych zmian</td> <td>K_K06</td> </tr> </table>	zna założenia metodyk zwinnych w odniesieniu do projektów programistycznych	K_W07, K_W08, Inż_W01	zna wybrane wzorce projektowe stosowane w rozwoju oprogramowania	K_W07, Inż_W01	zna podstawowe metody pracy z kodem obiektowo zorientowanym	K_W07, Inż_W01	potrafi posługiwać się zintegrowanym środowiskiem do rozwoju oprogramowania	K_U03, K_U07, Inż_U03, Inż_U05	potrafi napisać testy jednostkowe przy użyciu wybranej platformy testowej	K_U08, K_U10, K_U13, Inż_U01, Inż_U02	potrafi korzystać z wybranych elementów języka C++ w praktyce	K_U03, K_U07, Inż_U04	ma świadomość potrzeby dzielenia się wiedzą w ramach zespołu projektowego	K_K06	rozumie potrzebę dbania o jakość oprogramowania w kontekście pracy zespołowej	K_K06	rozumie konieczność stałego dokształcania się w obliczu ciągłych zmian	K_K06
zna założenia metodyk zwinnych w odniesieniu do projektów programistycznych	K_W07, K_W08, Inż_W01																		
zna wybrane wzorce projektowe stosowane w rozwoju oprogramowania	K_W07, Inż_W01																		
zna podstawowe metody pracy z kodem obiektowo zorientowanym	K_W07, Inż_W01																		
potrafi posługiwać się zintegrowanym środowiskiem do rozwoju oprogramowania	K_U03, K_U07, Inż_U03, Inż_U05																		
potrafi napisać testy jednostkowe przy użyciu wybranej platformy testowej	K_U08, K_U10, K_U13, Inż_U01, Inż_U02																		
potrafi korzystać z wybranych elementów języka C++ w praktyce	K_U03, K_U07, Inż_U04																		
ma świadomość potrzeby dzielenia się wiedzą w ramach zespołu projektowego	K_K06																		
rozumie potrzebę dbania o jakość oprogramowania w kontekście pracy zespołowej	K_K06																		
rozumie konieczność stałego dokształcania się w obliczu ciągłych zmian	K_K06																		
16.	<p>Literatura obowiązkowa i zalecana</p> <p>Literatura podstawowa</p> <ul style="list-style-type: none"> • Stephen Prata – Język C++. Szkoła programowania. • Scott Meyers – Skuteczny, nowoczesny C++. 42 sposoby lepszego posługiwania się językami C++11 i C++14. • Kent Beck – TDD. Sztuka tworzenia dobrego kodu. • Robert C. Martin – Czysty kod. Podręcznik dobrego programisty. • Zasoby sieci Internet w odniesieniu do przedstawianych tematów. 																		
17.	<p>Metody weryfikacji zakładanych efektów kształcenia</p> <p>kolokwium, prezentacja napisanych programów komputerowych</p>																		
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Zaliczenie kursu odbywa się na podstawie obecności oraz wyniku kolokwium. Szczególna aktywność na zajęciach praktycznych jest dodatkowo premiowana.</p>																		
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="228 1417 1428 1485"> <tr> <td>pracownia</td> <td>30 godz.</td> </tr> <tr> <td>wykład</td> <td>15 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="228 1563 1428 1608"> <tr> <td>rozwiązywanie zadań</td> <td>25 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="228 1686 1428 1749"> <tr> <td>Łączna liczba godzin</td> <td>70 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>3</td> </tr> </table>	pracownia	30 godz.	wykład	15 godz.	rozwiązywanie zadań	25 godz.	Łączna liczba godzin	70 godz.	Liczba punktów ECTS	3								
pracownia	30 godz.																		
wykład	15 godz.																		
rozwiązywanie zadań	25 godz.																		
Łączna liczba godzin	70 godz.																		
Liczba punktów ECTS	3																		

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs: Tworzenie aplikacji frontendowych Front-end apps development
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DKFWebDev
6.	Rodzaj przedmiotu/modułu Kurs inżynierski (Kinż)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: <ul style="list-style-type: none"> • Wstęp do programowania w języku C lub Python Niezbędne kompetencje: <ul style="list-style-type: none"> • podstawowa umiejętność programowania • podstawowa znajomość dowolnego systemu kontroli wersji
13.	Cele kształcenia dla przedmiotu Wykład ma na celu przybliżyć słuchaczom rolę frontend developera i związane z nią odpowiedzialności, a także przekazać wiedzę umożliwiającą wytworzenie pierwszych frontendowych aplikacji.

14.	<p>Treści programowe</p> <ol style="list-style-type: none"> 1. Podstawy działania Internetu 2. HTML <ul style="list-style-type: none"> • Podstawy HTML • Dobre praktyki i konwencje • Formularze i walidacja 3. CSS <ul style="list-style-type: none"> • Podstawy CSS • Tworzenie Layoutu: Float, Grid, Flexbox, Positioning, Box Model • Responsywny design • Preprocesory CSS • BEM 4. JavaScript <ul style="list-style-type: none"> • Podstawy JavaScriptu • Manipulacja DOM • ES6+ • Zaawansowane pojęcia: Hoisting, Event Bubbling, Scope, Prototype, Shadow DOM • Event Loop • TypeScript 5. Komunikacja z Backendem <ul style="list-style-type: none"> • REST, SOAP?, GraphQL? 6. Menadżer Pakietów 7. Narzędzia <ul style="list-style-type: none"> • Bundlery: Webpack, EsBuild • NPM • Lintery i formatery 8. Frameworki/Biblioteki <ul style="list-style-type: none"> • Przegląd dostępnych opcji • Podstawy React'a 9. Nowoczesny CSS <ul style="list-style-type: none"> • Styled Components • CSS Modules • Frameworki CSS: Material, Bootstrap, inne? 10. Testowanie Aplikacji 11. Optymalizacja 																								
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">Posiada przeglądową wiedzę na temat wytwarzania aplikacji frontendowych</td> <td style="padding: 5px;">K_W07, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">Zna różne metody wytwarzania nowoczesnych aplikacji frontendowych</td> <td style="padding: 5px;">K_W08, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">Zna różne metody konfigurowania nowoczesnego środowiska developerskiego</td> <td style="padding: 5px;">K_W08, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">Zna podstawowe konstrukcje programistyczne oraz standardowe typy danych w języku JavaScript</td> <td style="padding: 5px;">K_W05, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">Ma podstawową wiedzę na temat nowoczesnych Frameworków używanych w codziennej pracy Frontend Developera</td> <td style="padding: 5px;">K_W08, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">Zna podstawy tworzenia różnego rodzaju rozwiązań w Internecie</td> <td style="padding: 5px;">K_W07, Inż_W01</td> </tr> </table> <p>Umiejętności</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">Potrafi podążać za zmianami technologicznymi</td> <td style="padding: 5px;">Inż_U04</td> </tr> <tr> <td style="padding: 5px;">Umie tworzyć rozwiązania frontendowe w oparciu o dostępne zasoby (narzędzia, biblioteki, frameworki itd.)</td> <td style="padding: 5px;">K_U03, K_U07, K_U08, K_U10, Inż_U02, Inż_U03</td> </tr> <tr> <td style="padding: 5px;">Umie wykorzystać istniejące zasoby frontendowe w budowie własnego rozwiązania</td> <td style="padding: 5px;">K_U07, K_U09, Inż_U02, Inż_U03</td> </tr> <tr> <td style="padding: 5px;">Umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy</td> <td style="padding: 5px;">K_U12</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">Rozumie z czym wiąże się praca Frontend Developera</td> <td style="padding: 5px;">K_K02, K_K06</td> </tr> <tr> <td style="padding: 5px;">Rozumie znaczenie roli Frontend Developera</td> <td style="padding: 5px;">K_K02, K_K06</td> </tr> </table>	Posiada przeglądową wiedzę na temat wytwarzania aplikacji frontendowych	K_W07, Inż_W01	Zna różne metody wytwarzania nowoczesnych aplikacji frontendowych	K_W08, Inż_W01	Zna różne metody konfigurowania nowoczesnego środowiska developerskiego	K_W08, Inż_W01	Zna podstawowe konstrukcje programistyczne oraz standardowe typy danych w języku JavaScript	K_W05, Inż_W01	Ma podstawową wiedzę na temat nowoczesnych Frameworków używanych w codziennej pracy Frontend Developera	K_W08, Inż_W01	Zna podstawy tworzenia różnego rodzaju rozwiązań w Internecie	K_W07, Inż_W01	Potrafi podążać za zmianami technologicznymi	Inż_U04	Umie tworzyć rozwiązania frontendowe w oparciu o dostępne zasoby (narzędzia, biblioteki, frameworki itd.)	K_U03, K_U07, K_U08, K_U10, Inż_U02, Inż_U03	Umie wykorzystać istniejące zasoby frontendowe w budowie własnego rozwiązania	K_U07, K_U09, Inż_U02, Inż_U03	Umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy	K_U12	Rozumie z czym wiąże się praca Frontend Developera	K_K02, K_K06	Rozumie znaczenie roli Frontend Developera	K_K02, K_K06
Posiada przeglądową wiedzę na temat wytwarzania aplikacji frontendowych	K_W07, Inż_W01																								
Zna różne metody wytwarzania nowoczesnych aplikacji frontendowych	K_W08, Inż_W01																								
Zna różne metody konfigurowania nowoczesnego środowiska developerskiego	K_W08, Inż_W01																								
Zna podstawowe konstrukcje programistyczne oraz standardowe typy danych w języku JavaScript	K_W05, Inż_W01																								
Ma podstawową wiedzę na temat nowoczesnych Frameworków używanych w codziennej pracy Frontend Developera	K_W08, Inż_W01																								
Zna podstawy tworzenia różnego rodzaju rozwiązań w Internecie	K_W07, Inż_W01																								
Potrafi podążać za zmianami technologicznymi	Inż_U04																								
Umie tworzyć rozwiązania frontendowe w oparciu o dostępne zasoby (narzędzia, biblioteki, frameworki itd.)	K_U03, K_U07, K_U08, K_U10, Inż_U02, Inż_U03																								
Umie wykorzystać istniejące zasoby frontendowe w budowie własnego rozwiązania	K_U07, K_U09, Inż_U02, Inż_U03																								
Umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy	K_U12																								
Rozumie z czym wiąże się praca Frontend Developera	K_K02, K_K06																								
Rozumie znaczenie roli Frontend Developera	K_K02, K_K06																								

16.	Literatura obowiązkowa i zalecana <ul style="list-style-type: none"> • https://roadmap.sh/frontend 	
17.	Metody weryfikacji zakładanych efektów uczenia się Prezentacja rozwiązania zadań	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Zdobycie wymaganej liczby punktów z pracowni	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	30 godz.
	pracownia	30 godz.
	Praca własna studenta	
	Rozwiązywanie zadań na pracownię	40 godz.
	Studiowanie materiału z wykładu	20 godz.
	Sumarycznie	
	Łączna liczba godzin	120 godz.
	Liczba punktów ECTS	5

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs administrowania systemem Linux Linux system administration
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DKASL
6.	Rodzaj przedmiotu/modułu Kurs inżynierski (Kinż)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Brak
13.	Cele kształcenia dla przedmiotu Celem kursu jest rozwinięcie u uczestników praktycznych umiejętności zarządzania systemem Linux na komputerze osobistym lub serwerze. Zostaną przedstawione podstawowe informacje na temat budowy i sposobu działania systemów uniksopodobnych, a szczególnie różnych dystrybucji Linuksa. Zajęcia są adresowane do wszystkich studentów, którzy chcieliby sprawnie korzystać na co dzień z systemów linuksowych.

I. WSTĘP: ŻYCIE Z LINUKSEM

1. Hardware i software. Budowa komputera i struktura oprogramowania podstawowego. Jądro i przestrzeń użytkownika. Proces rozruchu komputera. Instalacja systemu i rozpoczęcie pracy z Linuksem.
2. Terminal tekstowy. Powłoka systemowa. Podstawowe polecenia. Uruchamianie programów, standardowe strumienie wejściowe i wyjściowe, potoki. Uruchamianie i zatrzymywanie procesów. Zarządzanie plikami, prawa dostępu, kopiowanie, archiwizowanie. Edytory tekstu. Przetwarzanie plików tekstowych.
3. Użytkownik root. Hasła. Su i sudo.

II. ZAGADNIENIA SZCZEGÓŁOWE

1. Hardware i jego diagnostyka, lshw, lsusb, lspci, dmidecode, i2c-tools i in. Urządzenia. Sysfs i udev. Udevd i udevadm.
2. Dyski. Protokoły SCSI i ATA. USB Storage. Hdparm i smartmontools.
3. Partycje. MBR i GPT. Narzędzia do partycjonowania dysków, fdisk, parted.
4. Urządzenia blokowe, device mapper, losetup, dm-crypt, LVM.
5. Systemy plików. Budowa systemu plików, inode. Księgowanie. Systemy ext2, ext3 i ext4. Tworzenie i zarządzanie systemami plików. Diagnostyka systemów plików, fsck. Montowanie, mount, /etc/fstab itp. Dowiązania, pliki specjalne. FAT i NTFS. Filesystem in userspace. Systemy plików btrfs, ufs, zfs, ffs i in. Szczególne wymagania dysków SSD, discard i fstrim. Linux Directory Structure, katalogi /etc, /usr, /var, /tmp, /boot i ich przeznaczenie.
6. Przestrzeń wymiany w osobnej partycji i w pliku. Przestrzeń wymiany w pamięci RAM, zram, zswap.
7. Rozruch komputera. Firmware, Legacy BIOS i UEFI. Konfiguracja, efibootmgr. Inne (Coreboot, Libreboot, U-Boot). Bootloadery drugiego poziomu. GNU Grub 2 i jego konfiguracja. Informacja o Legacy GRUB. Inne popularne bootloadery, Syslinux, LILO, systemd-boot (Gummiboot). Konfiguracja wielosystemowa z BIOS i UEFI, współżycie z systemem MS Windows, rEFInd. PXE i rozruch poprzez sieć komputerową. BOOTP i TFTP. Plymouth i estetyka rozruchu.
8. Start jądra. Tworzenie i konfigurowanie initramfs. Uruchamianie awaryjne i tryb pojedynczego użytkownika. Moduły jądra, modprobe, lsmod, rmmmod, /etc/modules.
9. Konfiguracja systemu. Katalog /etc i jego struktura.
10. Uruchamianie przestrzeni użytkownika. System V Init. Proces init, inittab, init.d i rc?.d, runlevels, telinit, update-rc.d, insserv, run-parts. Informacja o Upstart.
11. Systemd i jego konfiguracja. Filozofia działania systemd. Organizacja i składnia plików konfiguracyjnych. Demony systemd. Jednostki systemd. Cele i metody ich osiągania. Serwisy i inne jednostki.
12. Zarządzanie użytkownikami. Logowanie do systemu, getty i login. Konsola zdalna i minicom. Hasła, passwd i shadow. PAM. Własność procesów, suid.
13. Dziennik systemowy, rsyslog, logger, /var/log, journalctl i dmesg. Zdalny zapis dziennika. Protokół SMTP i narzędzia do korzystania z niego.
14. Synchronizacja czasu, NTP, ntpd, ntpdate, open-ntp, chrony, systemd-timesyncd.
15. Planowanie wykonania procesów, procesy wsadowe. Cron, crontab, anacron, systemd-timer. Polecenie at.
16. Zarządzanie procesami i ich diagnostyka, ps, lsof, strace, ltrace. Wysyłanie sygnałów do procesów, kill, killall, nohup. Wątki. Pomiar czasu procesora, obciążenia, zużycia pamięci i operacji wejścia/wyjścia (top, times, vmstat, iostat, iotop, pidstat itp). Priorytety procesów, nice. Komunikacja międzyprocesowa. Gniazda. Dbus. Komunikacja między użytkownikami, wall.
17. Biblioteki współdzielone, /lib, /usr/lib, ldd, ldconfig, /etc/ld.so.conf, LD_LIBRARY_PATH.
18. Repozytoria pakietów i dystrybucje Linuksa. Debian, Ubuntu i pakiety deb. Konfigurowanie apt i aptitude. RHEL, Fedora, Centos i rpm. Arch Linux i pacman. Dystrybucje specjalizowane, Kali Linux, Tails i in. Krzyżowanie dystrybucji. Linux from scratch.
19. Efektywna praca w trybie tekstowym. Powłoki bash (sh, dash, ash), c-shell (csh, tcsh), zsh. Biblioteka readline. Terminal, terminfo, termcap. Screen i tmux. Konfiguracja interpretera powłoki. Tekst zachęty, Powershell. Lokalizacja i internacjonalizacja, locale, tzdata, kodowanie znaków.
20. Skrypty powłoki. Narzędzia sed, awk. Polecenia użytkowe wc, tr, sort, head, tail, tee, xargs, grep, find, locate i wiele innych.
21. Separacja procesów, namespaces, cgroups. Piaskownice, firejail i in. Apparmor. Selinux.
22. Wirtualizacja. LXC i docker. Qemu i KVM. Xen. Informacja o programie Virtualbox.
23. Sieci komputerowe i podstawowa konfiguracja sieci w Linuksie. Podstawowe polecenia konfiguracyjne. Konfiguracja statyczna i dynamiczna. DHCP. Gniazda. Bezpieczeństwo i iptables. Usługi sieciowe, sshd, inetd i xinetd. Gniazda systemd. RPC. Usługi nazw, bind9.
24. Usługi drukowania, lpr i Cups.

Zakres materiału obejmuje większość zagadnień wymaganych na egzaminach certyfikacyjnych CompTIA Linux+ (LX0-103 i LX0-104) oraz LPIC-1 (LPI 101 i LPI 102 — Certified Linux Administrator).

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 192 1153 286"> <tr> <td>Zna podstawowe funkcje systemu operacyjnego i mechanizmy jego działania</td> <td>K_W07, K_W08, Inż_W01</td> </tr> <tr> <td>Zna podstawowe narzędzia administracyjne w Linuksie</td> <td>K_W08, Inż_W01</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 376 1153 499"> <tr> <td>Potrafi efektywnie wykonywać podstawowe czynności administracyjne w Linuksie</td> <td>K_U07, Inż_U04, Inż_U05</td> </tr> <tr> <td>Potrafi diagnozować problemy powstające podczas pracy systemu operacyjnego i sprawnie je rozwiązywać</td> <td>K_U09, Inż_U01, Inż_U02, Inż_U05</td> </tr> </table>	Zna podstawowe funkcje systemu operacyjnego i mechanizmy jego działania	K_W07, K_W08, Inż_W01	Zna podstawowe narzędzia administracyjne w Linuksie	K_W08, Inż_W01	Potrafi efektywnie wykonywać podstawowe czynności administracyjne w Linuksie	K_U07, Inż_U04, Inż_U05	Potrafi diagnozować problemy powstające podczas pracy systemu operacyjnego i sprawnie je rozwiązywać	K_U09, Inż_U01, Inż_U02, Inż_U05				
Zna podstawowe funkcje systemu operacyjnego i mechanizmy jego działania	K_W07, K_W08, Inż_W01												
Zna podstawowe narzędzia administracyjne w Linuksie	K_W08, Inż_W01												
Potrafi efektywnie wykonywać podstawowe czynności administracyjne w Linuksie	K_U07, Inż_U04, Inż_U05												
Potrafi diagnozować problemy powstające podczas pracy systemu operacyjnego i sprawnie je rozwiązywać	K_U09, Inż_U01, Inż_U02, Inż_U05												
16.	<p>Literatura obowiązkowa i zalecana</p> <ol style="list-style-type: none"> Brian Ward, Jak działa Linux. Podręcznik administratora, wydanie II, Helion 2015. Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley, Dan Mackin, Unix i Linux. Przewodnik administratora systemów, wydanie V, Helion 2018. 												
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <ul style="list-style-type: none"> prezentacja rozwiązań zadań przy komputerze 												
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Ocena zajęć zależy od liczby punktów za prezentację rozwiązań zadań zdobytych podczas zajęć na pracowni.</p>												
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 969 1425 1032"> <tr> <td>wykład</td> <td>30 godz.</td> </tr> <tr> <td>pracownia</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 1115 1425 1178"> <tr> <td>studiowanie tematyki wykładów i literatury</td> <td>30 godz.</td> </tr> <tr> <td>przygotowanie do pracowni</td> <td>30 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 1261 1425 1323"> <tr> <td>Łączna liczba godzin</td> <td>120 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>5</td> </tr> </table>	wykład	30 godz.	pracownia	30 godz.	studiowanie tematyki wykładów i literatury	30 godz.	przygotowanie do pracowni	30 godz.	Łączna liczba godzin	120 godz.	Liczba punktów ECTS	5
wykład	30 godz.												
pracownia	30 godz.												
studiowanie tematyki wykładów i literatury	30 godz.												
przygotowanie do pracowni	30 godz.												
Łączna liczba godzin	120 godz.												
Liczba punktów ECTS	5												

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Podstawy elektroniki, elektrotechniki i miernictwa Fundamentals of electronics, electrical engineering and metrology
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DOPEEM
6.	Rodzaj przedmiotu/modułu Kurs inżynierski (Kinż)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład – 30 godzin, pracownia – 20 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • Analiza matematyczna
13.	Cele przedmiotu <p>Celem wykładu jest zapoznanie studentów z wielkościami fizycznymi, które pozwalają na opisywanie i analizę układów elektronicznych analogowych oraz cyfrowych. Wiedza zdobyta na wykładzie pozwoli zrozumieć ograniczenia wynikające z właściwości fizycznych układów, które mają wpływ na np.: czas dostępu do zasobów, czas propagacji sygnałów w układach cyfrowych. Ponadto zrozumienie zagadnień opisujących zjawiska fizyczne pozwoli studentom na świadome i bezpieczne użytkowanie urządzeń elektronicznych. Odnosi się to zarówno do bezpieczeństwa samych użytkowników (napięcie dotykowe, zabezpieczenia różnicowoprądowe czy nadprądowe), jak i sprzętu elektronicznego (stany nieustalone, „przeładowanie” bramki).</p> <p>Umiejętności nabyte na pracowni pozwolą studentom na samodzielną realizację prostych układów elektronicznych i na świadome wykonanie pomiarów za pomocą multimetrów oraz oscyloskopów. Uczestnicy poznają budowę i zasadę działania podzespołów elektronicznych, w tym elementów półprzewodnikowych oraz scalonych wzmacniaczy operacyjnych.</p> <p>Na wykładzie studenci zapoznają się praktycznymi metodami analizy układów elektrycznych, które pozwalają na szybkie szacowanie wartości w oparciu o prawa: Ohma, Kirchhoffa oraz Thevenina. Studenci będą potrafili skorzystać z not aplikacyjnych układów o niewielkim stopniu skomplikowania, takich jak tranzystory i wzmacniacze operacyjne.</p>
14.	Treści programowe 1. podstawowe wielkości elektryczne i prawa opisujące relacje pomiędzy nimi, 2. obwody elektryczne (budowa i ich analiza), 3. obwody prądu przemiennego (energia i moc), 4. urządzenia elektryczne (silniki DC i AC), 5. zabezpieczenia obwodów 1-fazowych, 6. podstawowe przyrządy półprzewodnikowe (diody, tranzystory), 7. filtry, prostowniki, stabilizatory napięcia, 8. przetwornice napięcia, 9. wzmacniacze operacyjne i ich układy pracy, 10. przyrządy do pomiaru wielkości elektrycznych.

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <p>Ma wiedzę z zakresu podstaw elektroniki niezbędną do bezpiecznego podłączania i użytkowania urządzeń elektrycznych Rozumie na poziomie podstawowym działanie najpopularniejszych podzespołów elektronicznych Posiada wiedzę lub potrafi ją uzyskać z not katalogowych podstawowych komponentów elektronicznych: tranzystorów, wzmacniaczy, stabilizatorów lub przetwornic napięcia</p> <p>Umiejętności</p> <p>Potrafi podłączyć, zmierzyć i przeanalizować pracę prostych układów elektronicznych zgodnie z wytycznymi Potrafi skorzystać z informacji zawartych w notach aplikacyjnych podzespołów elektronicznych i na ich podstawie zrealizować układ</p> <p>Kompetencje społeczne</p> <p>Rozumie znaczenie zasad stosowanych przy podłączaniu i uruchamianiu sprzętu elektronicznego i elektrycznego dla bezpieczeństwa ludzi i sprzętu Jest świadomy ograniczeń wynikających z natury zjawisk fizycznych na parametry pracy urządzeń elektronicznych</p>	<p>K_W08</p> <p>Inż_W01</p> <p>Inż_W01</p> <p>K_U10, Inż_U01-03</p> <p>K_U13, Inż_U01-02</p> <p>K_K02</p> <p>K_K05</p>																
16.	<p>Literatura obowiązkowa i zalecana</p> <p>Obowiązkowa:</p> <ul style="list-style-type: none"> • Electronics Fundamentals: Circuits, Devices amp; Applications. Thomas L. Floyd, David M. Buchla, Pearson Education 2013 • Elektrotechnika, S. Bolkowski WSiP Warszawa 2005 Electronic Devices (Conventional Current Version), Thomas L. Floyd, Pearson Education <p>Zalecana:</p> <ul style="list-style-type: none"> • Sztuka elektroniki, P. Horowitz, W. Hill, WKŁ, Warszawa 1992, 1995. 																	
17.	<p>Metody weryfikacji zakładanych efektów kształcenia</p> <p>egzamin, sprawozdania/raporty</p>																	
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <ul style="list-style-type: none"> • Wykład – egzamin ustny, obserwacja aktywności studentów podczas wykładu; • Pracownia – realizacja zadań na zajęciach, obserwacje i ocena umiejętności praktycznych, opracowywanie wyników pomiarów, 																	
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 1532 1275 1599"> <tr> <td>wykład</td> <td>30 godz.</td> </tr> <tr> <td>pracownia</td> <td>20 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 1677 1275 1805"> <tr> <td>studiowanie tematyki wykładów i literatury</td> <td>20 godz.</td> </tr> <tr> <td>opracowywanie wyników</td> <td>15 godz.</td> </tr> <tr> <td>przygotowanie do egzaminu</td> <td>10 godz.</td> </tr> <tr> <td>przygotowanie do pracowni</td> <td>15 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 1890 1275 1953"> <tr> <td>Łączna liczba godzin</td> <td>110 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>4</td> </tr> </table>	wykład	30 godz.	pracownia	20 godz.	studiowanie tematyki wykładów i literatury	20 godz.	opracowywanie wyników	15 godz.	przygotowanie do egzaminu	10 godz.	przygotowanie do pracowni	15 godz.	Łączna liczba godzin	110 godz.	Liczba punktów ECTS	4	
wykład	30 godz.																	
pracownia	20 godz.																	
studiowanie tematyki wykładów i literatury	20 godz.																	
opracowywanie wyników	15 godz.																	
przygotowanie do egzaminu	10 godz.																	
przygotowanie do pracowni	15 godz.																	
Łączna liczba godzin	110 godz.																	
Liczba punktów ECTS	4																	

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Scala in Practice Scala in Practice
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-ScalaPr
6.	Rodzaj przedmiotu/modułu Kurs inżynierski (Kinż)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów ((jeśli obowiązuje)) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 30 godzin, pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Entry Recommendations: <ul style="list-style-type: none"> • Java, • Object Oriented Programming • basics of Web Development.
13.	Cele przedmiotu The goal of this course is to introduce students to the Scala language, along with the frameworks and libraries associated with it, which are a standard in the industry.

14.	<p>Treści programowe</p> <p>Program:</p> <p>I. Scala syntax</p> <ul style="list-style-type: none"> • Types • Classes and objects • Traits • Functions and closures • Collections • Case classes and pattern matching • Lambdas • Implicit parameters • Code standards <p>II. Frameworks and libraries</p> <ul style="list-style-type: none"> • Build tool - [Sbt] • Testing - [ScalaTest] • Database access - [Slick] • Web applications - [Play Framework] • Concurrent and distributed applications [Akka] • Functional Programming - [Cats] <p>III. Glimpse into the future of Scala [Dotty compiler]</p>												
15.	<p>Zakładane efekty uczenia się</p> <p>Knowledge</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">knows the syntax of Scala language</td> <td style="padding: 5px;">K_W05, K_W08, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">knows basics of object-oriented and functional programming</td> <td style="padding: 5px;">K_W05, K_W08, Inż_W01</td> </tr> <tr> <td style="padding: 5px;">has overview of standards and frameworks used in the industry</td> <td style="padding: 5px;">K_W07, Inż_W01</td> </tr> </table> <p>Skills</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">can implement a medium size web-based application (Play & Akka)</td> <td style="padding: 5px;">K_U03, K_U08, K_U13, Inż_U01, Inż_U02, Inż_U03, Inż_U05</td> </tr> <tr> <td style="padding: 5px;">can write unit-tests (ScalaTest)</td> <td style="padding: 5px;">K_U07, K_U10, Inż_U01, Inż_U02, Inż_U04, Inż_U05</td> </tr> <tr> <td style="padding: 5px;">can use build tool (Sbt)</td> <td style="padding: 5px;">K_U07, K_U10, Inż_U02, Inż_U04</td> </tr> </table>	knows the syntax of Scala language	K_W05, K_W08, Inż_W01	knows basics of object-oriented and functional programming	K_W05, K_W08, Inż_W01	has overview of standards and frameworks used in the industry	K_W07, Inż_W01	can implement a medium size web-based application (Play & Akka)	K_U03, K_U08, K_U13, Inż_U01, Inż_U02, Inż_U03, Inż_U05	can write unit-tests (ScalaTest)	K_U07, K_U10, Inż_U01, Inż_U02, Inż_U04, Inż_U05	can use build tool (Sbt)	K_U07, K_U10, Inż_U02, Inż_U04
knows the syntax of Scala language	K_W05, K_W08, Inż_W01												
knows basics of object-oriented and functional programming	K_W05, K_W08, Inż_W01												
has overview of standards and frameworks used in the industry	K_W07, Inż_W01												
can implement a medium size web-based application (Play & Akka)	K_U03, K_U08, K_U13, Inż_U01, Inż_U02, Inż_U03, Inż_U05												
can write unit-tests (ScalaTest)	K_U07, K_U10, Inż_U01, Inż_U02, Inż_U04, Inż_U05												
can use build tool (Sbt)	K_U07, K_U10, Inż_U02, Inż_U04												
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • Martin Odersky, Lex Spoon Bill Venners, Programming in Scala • Cay S. Horstmann, Scala for the Impatient • Alvin Alexander, Scala Cookbook: Recipes for Object-Oriented and Functional Programming • Raymond Roostenburg, Rob Bakker, and Rob Williams, Akka in Action 												
17.	<p>Metody weryfikacji zakładanych efektów kształcenia</p> <p>Programming exercises and demos.</p>												
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>To pass the course, one needs to gain the amount of points that are indicated in the regulations.</p>												

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 h
	wykład	30 h
	Praca własna studenta	
	self study	20 h
	programming	45 h
	Sumarycznie	
	Łączna liczba godzin	125 godz.
	Liczba punktów ECTS	5

6 Proseminaria

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Proseminarium: Bezpieczeństwo i ochrona informacji Seminar: Information Safety and Security
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-BiOI
6.	Rodzaj przedmiotu/modułu Proseminarium
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin seminarium — 28 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu zainteresowanie tematyką bezpieczeństwa i ochrony informacji
13.	Cele kształcenia dla przedmiotu Celem seminarium jest przedstawienie wiedzy z zakresu bezpieczeństwa i ochrony informacji oraz przygotowanie słuchaczy do zgodnego z RODO wytwarzania, testowania i eksploatacji oprogramowania.
14.	Treści programowe <ul style="list-style-type: none"> • aspekty prawne bezpieczeństwa i ochrony informacji • bezpieczeństwo aplikacji webowych i mobilnych • certyfikaty, podpisy elektroniczne • bezpieczeństwo sieci i infrastruktury • bezpieczeństwo rozwiązań chmurowych i IoT • analiza i testowanie bezpieczeństwa • zapewnienie bezpieczeństwa i ochrony informacji w procesie wytwarzania oprogramowania • zapewnienie ciągłości działania • znaczenie wprowadzenia i utrzymywania polityki bezpieczeństwa • przegląd najnowszych ataków

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="228 192 1157 353"> <tr> <td>Zna najważniejsze zagadnienia dotyczące zagrożenia bezpieczeństwa działania oprogramowania: ataków i awarii</td> <td>K_W08</td> </tr> <tr> <td>Ma podstawową wiedzę dotyczącą prawnych aspektów pracy informatyka, ze szczególnym uwzględnieniem obowiązującego prawa autorskiego i zagadnień ochrony danych osobowych</td> <td>K_W10, Inż_W02</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="228 439 1157 658"> <tr> <td>Potrafi wykonać prostą analizę podatności systemu informatycznego na ataki i awarie</td> <td>K_U09</td> </tr> <tr> <td>Potrafi prezentować opracowane zagadnienia, formułować opinie, a także podejmować dyskusję i analizować problemy dotyczące zagrożeń systemów informatycznych związanych z atakami i awaryjnością</td> <td>K_U12</td> </tr> <tr> <td>Potrafi dokonać krytycznej analizy zgodności wytwarzania, testowania i eksploatacji oprogramowania z RODO</td> <td>Inż_U02</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="228 743 1157 992"> <tr> <td>Jest świadom ograniczeń własnej wiedzy i umiejętności w zakresie zagrożeń (ataków i awaryjności), zabezpieczania oraz przepisów prawa dotyczących systemów informatycznych. Rozumie konieczność doskonalenia swoich zawodowych kompetencji w tym zakresie</td> <td>K_K01</td> </tr> <tr> <td>Wykazuje gotowość do wypełniania społecznych zobowiązań wynikających z charakteru pracy typowej dla absolwentów kierunku informatyka, w szczególności w zakresie ciągłej analizy zagrożeń i zapewnienia bezpiecznego działania systemów informatycznych</td> <td>K_K03</td> </tr> </table>	Zna najważniejsze zagadnienia dotyczące zagrożenia bezpieczeństwa działania oprogramowania: ataków i awarii	K_W08	Ma podstawową wiedzę dotyczącą prawnych aspektów pracy informatyka, ze szczególnym uwzględnieniem obowiązującego prawa autorskiego i zagadnień ochrony danych osobowych	K_W10, Inż_W02	Potrafi wykonać prostą analizę podatności systemu informatycznego na ataki i awarie	K_U09	Potrafi prezentować opracowane zagadnienia, formułować opinie, a także podejmować dyskusję i analizować problemy dotyczące zagrożeń systemów informatycznych związanych z atakami i awaryjnością	K_U12	Potrafi dokonać krytycznej analizy zgodności wytwarzania, testowania i eksploatacji oprogramowania z RODO	Inż_U02	Jest świadom ograniczeń własnej wiedzy i umiejętności w zakresie zagrożeń (ataków i awaryjności), zabezpieczania oraz przepisów prawa dotyczących systemów informatycznych. Rozumie konieczność doskonalenia swoich zawodowych kompetencji w tym zakresie	K_K01	Wykazuje gotowość do wypełniania społecznych zobowiązań wynikających z charakteru pracy typowej dla absolwentów kierunku informatyka, w szczególności w zakresie ciągłej analizy zagrożeń i zapewnienia bezpiecznego działania systemów informatycznych	K_K03
Zna najważniejsze zagadnienia dotyczące zagrożenia bezpieczeństwa działania oprogramowania: ataków i awarii	K_W08														
Ma podstawową wiedzę dotyczącą prawnych aspektów pracy informatyka, ze szczególnym uwzględnieniem obowiązującego prawa autorskiego i zagadnień ochrony danych osobowych	K_W10, Inż_W02														
Potrafi wykonać prostą analizę podatności systemu informatycznego na ataki i awarie	K_U09														
Potrafi prezentować opracowane zagadnienia, formułować opinie, a także podejmować dyskusję i analizować problemy dotyczące zagrożeń systemów informatycznych związanych z atakami i awaryjnością	K_U12														
Potrafi dokonać krytycznej analizy zgodności wytwarzania, testowania i eksploatacji oprogramowania z RODO	Inż_U02														
Jest świadom ograniczeń własnej wiedzy i umiejętności w zakresie zagrożeń (ataków i awaryjności), zabezpieczania oraz przepisów prawa dotyczących systemów informatycznych. Rozumie konieczność doskonalenia swoich zawodowych kompetencji w tym zakresie	K_K01														
Wykazuje gotowość do wypełniania społecznych zobowiązań wynikających z charakteru pracy typowej dla absolwentów kierunku informatyka, w szczególności w zakresie ciągłej analizy zagrożeń i zapewnienia bezpiecznego działania systemów informatycznych	K_K03														
16.	<p>Literatura obowiązkowa i zalecana</p> <ol style="list-style-type: none"> 1. Stallings W., Brown. : Bezpieczeństwo systemów informatycznych. Zasady i praktyka 2. Liderman K.: Bezpieczeństwo informacyjne 3. Depo J., Piwowarski J.: Bezpieczeństwo informacyjne 4. Stevens W.R.: Biblia TCP/IP 5. Weidman G. : Bezpieczny system w praktyce. Wyższa szkoła hackingu i testy penetracyjne 6. Wilhelm T.: Profesjonalne testy penetracyjne. Zbuduj własne środowisko do testów 														
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>Opracowanie i przedstawienie prezentacji na zadany temat.</p>														
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Ocena wygłoszonych referatów (wykładów audiowizualnych) pod kątem poprawności merytorycznej i komunikatywności przekazu. Konieczne jest również aktywne uczestnictwo w dyskusji na zajęciach.</p>														
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="228 1563 1428 1597"> <tr> <td>seminarium</td> <td>28 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="228 1675 1428 1742"> <tr> <td>studiowanie materiałów i opracowanie koncepcji prezentacji</td> <td>25 godz.</td> </tr> <tr> <td>przygotowanie prezentacji</td> <td>15 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="228 1821 1428 1892"> <tr> <td>Łączna liczba godzin</td> <td>68 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>3</td> </tr> </table>	seminarium	28 godz.	studiowanie materiałów i opracowanie koncepcji prezentacji	25 godz.	przygotowanie prezentacji	15 godz.	Łączna liczba godzin	68 godz.	Liczba punktów ECTS	3				
seminarium	28 godz.														
studiowanie materiałów i opracowanie koncepcji prezentacji	25 godz.														
przygotowanie prezentacji	15 godz.														
Łączna liczba godzin	68 godz.														
Liczba punktów ECTS	3														

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Proseminarium: Linux kawałek po kawałku Linux piece by piece
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DSLlinuxKPK
6.	Rodzaj przedmiotu/modułu Proseminarium
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin seminarium — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: <ul style="list-style-type: none"> • Kurs administrowania systemem Linux (nie wymagany, ale zalecany). Niezbędne kompetencje: <ul style="list-style-type: none"> • Znajomość podstaw administrowania systemem Linux. • Umiejętność korzystania z repozytoriów oprogramowania.
13.	Cele kształcenia dla przedmiotu Dogłębne poznanie architektury systemu linuksowego, jego komponentów i zależności pomiędzy nimi. Rozwinięcie umiejętności skompilowania ze źródeł, skompletowania i skonfigurowania działającego systemu linuksowego.

14.	<p>Treści programowe</p> <p>Aby stworzyć kompletny system operacyjny wykorzystujący jądro Linuksa potrzeba skompletować od autorów (upstream) kody źródłowe wielu programów, dostosować je (patch) oraz skompilować i skonfigurować kompletną przestrzeń użytkownika. W ten sposób powstają tzw. dystrybucje Linuksa. Aby korzystać z Linuksa, zwykle instalujemy jedną z takich dystrybucji, dzięki temu jesteśmy zwolnieni z konieczności kompilowania i konfigurowania systemu, a nawet nie musimy sobie zdawać sprawy z jego zawartości.</p> <p>Świetnym sposobem na poznanie architektury systemu linuksowego jest własnoręczne zbudowanie go krok po kroku od zera. Pomaga w tym projekt Gerarda Beekmansa „Linux from scratch” rozpoczęty w 1999 i nadal aktywnie rozwijany (obecna wersja: 11.1). Jest to podręcznik wzbogacony sporą liczbą pomocniczych materiałów (skrypty, patche itp.). Wykonanie kompletnego projektu LFS, poza głębokim poznaniem wewnętrznej struktury systemu linuksowego, pozwala na zdobycie umiejętności niezbędnych do samodzielnego budowania specjalnych wersji takich systemów przeznaczonych do specjalnych zastosowań.</p> <p>Każdy z uczestników zajęć otrzyma do opracowania fragment systemu LFS, który powinien dobrze zrozumieć i przygotować w domu, a następnie przedstawić wszystkim uczestnikom zajęć podczas dwugodzinnej prezentacji tak, by mogli oni odtworzyć omówiony fragment we własnych instalacjach. Na koniec semestru wszyscy uczestnicy zajęć powinni mieć gotowe, działające instalacje.</p>														
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 786 1423 976"> <tr> <td data-bbox="225 786 1166 880">Wie, z jakich komponentów składa się system linuksowy (bootloader, jądro, init system, demony itd.), jakie są ich zadania i jak je można dostosowywać do własnych potrzeb</td> <td data-bbox="1166 786 1423 880">K_W08</td> </tr> <tr> <td data-bbox="225 880 1166 976">Zna alternatywne rozwiązania różnych komponentów systemu linuksowego i potrafi wybrać najlepiej dostosowane do swoich potrzeb (np. OpenRC zamiast SystemD itp.)</td> <td data-bbox="1166 880 1423 976">K_W08</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 1061 1423 1285"> <tr> <td data-bbox="225 1061 1166 1093">Potrafi skonfigurować każdy z komponentów systemu linuksowego</td> <td data-bbox="1166 1061 1423 1093">K_U10</td> </tr> <tr> <td data-bbox="225 1093 1166 1184">Potrafi pobrać z repozytoriów źródłowych, skompilować i skonfigurować duże pakiety oprogramowania. W szczególności potrafi skompilować własną wersję jądra systemu Linux</td> <td data-bbox="1166 1093 1423 1184">K_U10</td> </tr> <tr> <td data-bbox="225 1184 1166 1216">Potrafi współpracować z innymi nad wspólnym projektem</td> <td data-bbox="1166 1184 1423 1216">K_U14</td> </tr> <tr> <td data-bbox="225 1216 1166 1285">Potrafi samodzielnie opracować zadany temat i przygotować na ten temat prezentację</td> <td data-bbox="1166 1216 1423 1285">K_U12</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 1368 1423 1431"> <tr> <td data-bbox="225 1368 1166 1431">Umie przedstawić pozostałym członkom grupy efekty własnej pracy, w szczególności zebrane informacje i własne doświadczenia</td> <td data-bbox="1166 1368 1423 1431">K_K05</td> </tr> </table>	Wie, z jakich komponentów składa się system linuksowy (bootloader, jądro, init system, demony itd.), jakie są ich zadania i jak je można dostosowywać do własnych potrzeb	K_W08	Zna alternatywne rozwiązania różnych komponentów systemu linuksowego i potrafi wybrać najlepiej dostosowane do swoich potrzeb (np. OpenRC zamiast SystemD itp.)	K_W08	Potrafi skonfigurować każdy z komponentów systemu linuksowego	K_U10	Potrafi pobrać z repozytoriów źródłowych, skompilować i skonfigurować duże pakiety oprogramowania. W szczególności potrafi skompilować własną wersję jądra systemu Linux	K_U10	Potrafi współpracować z innymi nad wspólnym projektem	K_U14	Potrafi samodzielnie opracować zadany temat i przygotować na ten temat prezentację	K_U12	Umie przedstawić pozostałym członkom grupy efekty własnej pracy, w szczególności zebrane informacje i własne doświadczenia	K_K05
Wie, z jakich komponentów składa się system linuksowy (bootloader, jądro, init system, demony itd.), jakie są ich zadania i jak je można dostosowywać do własnych potrzeb	K_W08														
Zna alternatywne rozwiązania różnych komponentów systemu linuksowego i potrafi wybrać najlepiej dostosowane do swoich potrzeb (np. OpenRC zamiast SystemD itp.)	K_W08														
Potrafi skonfigurować każdy z komponentów systemu linuksowego	K_U10														
Potrafi pobrać z repozytoriów źródłowych, skompilować i skonfigurować duże pakiety oprogramowania. W szczególności potrafi skompilować własną wersję jądra systemu Linux	K_U10														
Potrafi współpracować z innymi nad wspólnym projektem	K_U14														
Potrafi samodzielnie opracować zadany temat i przygotować na ten temat prezentację	K_U12														
Umie przedstawić pozostałym członkom grupy efekty własnej pracy, w szczególności zebrane informacje i własne doświadczenia	K_K05														
16.	<p>Literatura obowiązkowa i zalecana</p> <p>1. Linux From Scratch, created by Gerard Beekmans, managing editor: Bruce Dubbs, editors: Douglas R. Reno, DJ Lucas.</p>														
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>Opracowanie i przedstawienie prezentacji na zadany temat.</p>														
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Aby zaliczyć przedmiot należy:</p> <ul style="list-style-type: none"> • w trakcie całego semestru kompilować i konfigurować krok po kroku swój własny system; • uczestniczyć w zajęciach omawiających kolejne komponenty systemu i wykorzystywać zdobytą na nich wiedzę do tworzenia własnego systemu; • przygotować i przeprowadzić jedne z takich zajęć (prezentacja, pokaz na żywo); • na ostatnich zajęciach przedstawić swój gotowy, działający system. 														

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	przygotowanie prezentacji	35 godz.
	Sumarycznie	
	Łączna liczba godzin	65 godz.
Liczba punktów ECTS	3	

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Proseminarium: Testowanie oprogramowania Proseminar: Software Testing
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DPsTestPrg
6.	Rodzaj przedmiotu/modułu Proseminarium
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin seminarium – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu <ul style="list-style-type: none"> ● Znajomość języków programowania: C++, Javy lub Pythona. ● Zaliczenie przedmiotu Testowanie oprogramowania, ew. Testowanie gier lub aktualne zaliczenie tego przedmiotu. ● Zalecane Programowanie obiektowe.
13.	Cele przedmiotu Celem seminarium jest rozszerzenie wiedzy z zakresu testowania oprogramowania.
14.	Treści programowe Zapoznanie studentów z nowoczesnymi, praktycznymi metodami i narzędziami testowania dotyczącymi następujących obszarów: <ul style="list-style-type: none"> ● testy funkcjonalne, ● testy нефункционалне, ● testowanie oprogramowania zorientowanego obiektowo, ● testowanie oprogramowania internetowego, ● testowanie oprogramowania mobilnego, ● testowanie użyteczności, ● testowanie gier, ● testowanie integracyjne, ● testowanie systemowe, ● testowanie bezpieczeństwa, ● testowanie wydajności, ● strategię planowania testów, ● efektywne zarządzanie testami, ● zgodne z RODO testowanie oprogramowania.

15.	Zakładane efekty uczenia się	
Wiedza		
Zna wybrane metody, techniki i narzędzia testowania oprogramowania oraz podstawy ich funkcjonowania		K_W07
Ma wiedzę dotyczącą społecznych i etycznych aspektów pracy informatyka, ze szczególnym uwzględnieniem zapewnienia poprawnego działania oprogramowania		K_W07
Umiejętności		
Potrafi formułować opinie na temat procesu testowania oprogramowania, a także prowadzić dyskusję przedstawiając i oceniając różne opinie i stanowiska z tym związane		K_U12
Potrafi przygotować prezentacje dotyczące zaawansowanych zagadnień testowania oprogramowania i przedstawiać je osobom niebędącym specjalistami		K_U12
Potrafi samodzielnie zdobywać wiedzę i rozwijać umiejętności zawodowe w obszarze metod, techniki i procesu testowania oprogramowania		K_U13
Kompetencje społeczne		
Jest świadom możliwości popełniania błędów podczas programowania. Wykazuje rozważny krytycyzm wobec opinii na temat poprawnego działania oprogramowania		K_K01
Wykazuje gotowość do pełnienia, w sposób odpowiedzialny i respektujący zasady etyki zawodowej, roli testera oprogramowania		K_K04
Wykazuje samodzielność myślenia i działania przy rozwiązywaniu problemów i wykonywaniu zadań typowych dla testowania		K_K05
16.	Literatura obowiązkowa i zalecana	
<ol style="list-style-type: none"> 1. Roman A.: Testowanie i jakość oprogramowania, PWN 2015 2. Roman A., Zmitrowicz K. (red): Testowanie w praktyce. Studium przypadków, PWN 2016 3. Zmitrowicz K.: Jakość projektów informatycznych. Rozwój i testowanie oprogramowania, Helion 2016 4. Smiglin R.: Zawód tester, Helion 2015 5. Pawlak R.: Testowanie oprogramowania, podręcznik dla początkujących, Helion 2014 6. Copeland D. A Practitioner's Guide to Software Test Design, Artech House 2004 7. Crispin L., Gregory C.: Agile Testing: A Practical Guide for Testers and Agile Teams, 2009 		
17.	Metody weryfikacji zakładanych efektów kształcenia	
Opracowanie i przedstawienie prezentacji na zadany temat.		
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
Ocena wygłoszonych referatów (wykładów audiowizualnych) pod kątem poprawności merytorycznej i komunikatywności przekazu. Konieczne jest również aktywne uczestnictwo w dyskusji na zajęciach.		
19.	Nakład pracy studenta	
Zajęcia z udziałem nauczyciela		
seminarium		30 godz.
Praca własna studenta		
studiowanie materiałów i opracowanie koncepcji prezentacji		20 godz.
przygotowanie prezentacji		10 godz.
seminarium		30 godz.
Sumarycznie		
Łączna liczba godzin		90 godz.
Liczba punktów ECTS		3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Proseminarium: Wizualizacja danych Proseminar: Data Visualisation
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-PsWizDan
6.	Rodzaj przedmiotu/modułu Proseminarium
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin seminarium — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Brak
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest zapoznanie studentów z ponadczasowymi zasadami dotyczącymi prezentowania danych, a także z aktualnymi sposobami tworzenia wizualizacji.
14.	Treści programowe Przedmiot omawia podstawowe uniwersalne prawa dotyczące tworzenia wizualizacji, w tym dobór kolorów i formatu, a następnie przechodzi do wizualizowania różnego rodzaju danych: <ul style="list-style-type: none"> • wizualizacja słów (wordle), tekstów (w tym redukcja przestrzeni wielowymiarowych), • wizualizacja rzeczy i zdarzeń na mapach, w tym tworzenie map ciepła (heatmap), • wizualizacja danych społecznościowych i innych rodzajów grafów, • grafy przepływu, • i inne, w zależności od aktualnej mody. Oprócz tego na zajęciach będziemy omawiać współczesne narzędzia do wizualizacji i ich możliwości.

15.	Zakładane efekty uczenia się	
	Wiedza	
	Zna zasady tworzenia czytelnych wizualizacji danych	K_W08
	Zna podstawowe narzędzia do tworzenia wizualizacji	K_W08
	Umiejętności	
Potrafi samodzielnie zgłębić zadany temat i przygotować na ten temat prezentację	K_U12	
Potrafi wygłosić prezentację multimedialną oraz odpowiadać na zadawane w niej pytania, w tym potrafi zadbać o techniczną stronę prezentacji (stworzenie slajdów, odpowiednie ich odtworzenie, zadbanie o odpowiednie warunki do prezentowania narzędzi informatycznych w czasie swojej prezentacji)	K_U12	
Potrafi odróżnić dobrą wizualizację od złej	K_U10	
Potrafi odpowiednio dobrać typ wizualizacji do danych	K_U10	
Potrafi zaprogramować kilka rodzajów wizualizacji danych	K_U10	
Kompetencje społeczne		
Ma umiejętność prezentowania w przystępny sposób swojej wiedzy innym	K_K05	
16.	Literatura obowiązkowa i zalecana Steele, J., Iliinsky, N. (2010). Beautiful visualization: Looking at data through the eyes of experts. O'Reilly Media, Inc.	
17.	Metody weryfikacji zakładanych efektów uczenia się prezentacja materiału przez studenta, stworzenie wizualizacji przez studenta	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Student powinien przeprowadzić prezentację i stworzyć trzy wizualizacje używając technik omówionych na zajęciach. Ocena takiej osoby będzie średnią ważoną tych dwóch ocen, z wagami: <ul style="list-style-type: none"> • prezentacja 75% • wizualizacje 25% 	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	dotatkowe konsultacje w ramach potrzeb	4 godz.
	projekt końcowy	10 godz.
	przygotowywanie się do prezentacji (w tym czytanie materiałów dodatkowych)	35 godz.
Sumarycznie		
Łączna liczba godzin	79 godz.	
Liczba punktów ECTS	3	

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Proseminarium: Zaawansowane administrowanie systemem Linux Advanced Linux Administration
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-PSZASLnx
6.	Rodzaj przedmiotu/modułu Proseminarium
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin seminarium – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: <ul style="list-style-type: none"> • Kurs administrowania systemem Linux (nie wymgany, ale polecany). Niezbędne kompetencje: <ul style="list-style-type: none"> • znajomość budowy i działania systemu Linux, • umiejętność administrowania systemem Linux.
13.	Cele przedmiotu Na zajęciach będą omawiane różne zaawansowane zagadnienia administrowania systemami linuksowymi i ogólniej — uniksowymi. Zakłada się, że uczestnicy zajęć posiadli już pewną wiedzę, np. w zakresie Kursu administrowania Linuksem, choć wcześniejsze ukończenie tego kursu nie jest wymagane.

14.	<p>Treści programowe</p> <ul style="list-style-type: none"> • Dźwięk w Linuksie i Uniksie. Hardware: karty dźwiękowe. Ich obsługa w jądrze: Advanced Linux Sound Architecture (ALSA). Informacja o starszym systemie Open Sound System (OSS). Dźwięk we FreeBSD. Serwery dźwięku: PulseAudio i JACK. Wyższe warstwy: KDE Phonon, OpenAL. Oprogramowanie aplikacyjne: Audacity, SOX. Miksery i odtwarzacze (Xmms2, Amarok i in.). • Obraz w Linuksie. Terminale tekstowe. Framebuffer. Kernel Mode Setting. • Serwery obrazu. Architektura serwera XWindows. • Nowoczesna obsługa interfejsu graficznego: serwer obrazu Weyland. • Zarządcy obrazu, sesji, okien, pulpitu. Środowiska graficzne. Przegląd. Protokoły. Infrastruktura KDE i Gnome. • Advanced Configuration and Power Interface (ACPI). Zarządzanie stanami. Tabele. ACPI w Linuksie i FreeBSD. • Hardware płyty głównej. Unified Extensible Firmware Interface (UEFI). Tianocore. Coreboot. Środowiska uruchomieniowe x86_64 i arm64. • Sprzętowe wspomaganie grafiki. Direct Rendering Infrastructure (DRI). • Device mapper. Zastosowania: dm-crypt, dm-raid, dm-cache, dm-verity, LVM2 i in. • Systemy plików COW. Btrfs i ZFS. Migawki i kopie zapasowe. • Zaawansowane konfigurowanie sieci w Linuksie. QoS, traffic shaping i in. • Firewall i Intrusion Prevention Systems. Iptables i nftables. Konfigurowanie w Linuksie. SNORT. Filtrowanie pakietów we FreeBSD i OpenBSD: PF, IPFW, IPF. • Zdalne monitorowanie systemów. Protokoły SNMP i Syslog. Przegląd systemów: Nagios, Munin, Cacti, RRDTOol i in. • Zdalne konfigurowanie systemów. Systemy zarządzania konfiguracją: Ansible, Salt, Puppet i in. • Wirtualizacja. Xen. KVM. Narzędzia: Qemu, Virtualbox, Bhyve. Zarządzanie maszynami wirtualnymi: virt-manager, vagrant i in. • IaaS. Tworzenie chmur obliczeniowych. OpenStack, OpenNebula i in. • Zarządzanie klastrami. Kubernetes, Docker Swarm i in. <p>Uwagi</p> <ul style="list-style-type: none"> • Tematem zajęć może być wiele innych zagadnień, zależnie od zainteresowań słuchaczy (do uzgodnienia po rozpoczęciu zajęć). • Niektóre z powyższych zagadnień są dostatecznie obszerne, by być tematem więcej niż jednego spotkania. 						
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 1283 1423 1317"> <tr> <td>Ma zaawansowaną wiedzę na temat wybranych podsystemów Linuksa</td> <td>K_W07</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 1402 1423 1529"> <tr> <td>Umie zarządzać systemem Linux w stopniu zaawansowanym i wykorzystywać go do różnych potrzeb</td> <td>K_U10</td> </tr> <tr> <td>Potrafi przygotować prezentację dotyczącą zaawansowanych zagadnień z dziedziny administrowania systemem Linux</td> <td>K_U12</td> </tr> </table>	Ma zaawansowaną wiedzę na temat wybranych podsystemów Linuksa	K_W07	Umie zarządzać systemem Linux w stopniu zaawansowanym i wykorzystywać go do różnych potrzeb	K_U10	Potrafi przygotować prezentację dotyczącą zaawansowanych zagadnień z dziedziny administrowania systemem Linux	K_U12
Ma zaawansowaną wiedzę na temat wybranych podsystemów Linuksa	K_W07						
Umie zarządzać systemem Linux w stopniu zaawansowanym i wykorzystywać go do różnych potrzeb	K_U10						
Potrafi przygotować prezentację dotyczącą zaawansowanych zagadnień z dziedziny administrowania systemem Linux	K_U12						
16.	<p>Literatura obowiązkowa i zalecana</p> <p>Różna, zależnie od tematu prezentacji. W sporej części dokumentacja systemu Linux i jego komponentów (w tym jądra), artykuły specjalistyczne (np. z konferencji Usenix) itp.</p>						
17.	<p>Metody weryfikacji zakładanych efektów kształcenia</p> <p>Prezentacja, uczestnictwo w zajęciach.</p>						
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Ocena końcowa jest wyliczana na podstawie oceny otrzymanej za prezentację oraz liczby obecności na pozostałych zajęciach.</p>						

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	przygotowanie dwuniej prezentacji	35 godz.
	dodatkowe konsultacje na temat własnej prezentacji, zależności od potrzeb	0 godz.
	Sumarycznie	
	Łączna liczba godzin	65 godz.
	Liczba punktów ECTS	3

7 Projekty programistyczne

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt: boty konwersacyjne i odpowiadanie na pytania Project: creating a chatbot
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DPZBotKonw
6.	Rodzaj przedmiotu/modułu Projekt
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty <ol style="list-style-type: none">1. Algorytmy i struktury danych2. Przynajmniej jeden przedmiot z grupy: Sztuczna inteligencja, Machine Learning, Neural Networks and Natural Language Processing Inne wymagania <ol style="list-style-type: none">1. Umiejętność programowania w Pythonie2. Umiejętność pracy w grupie3. Umiejętność pracy z cudzym kodem
13.	Cele przedmiotu Podstawowym celem przedmiotu jest zapoznanie studentów ze współczesnymi metodami tworzenia agentów konwersacyjnych, w szczególności wykorzystujących techniki Deep Learningu.
14.	Treści programowe <ol style="list-style-type: none">1. Modele językowe jako narzędzia do tworzenia systemów dialogowych oraz odpowiadania na pytania.2. Podstawy tłumaczenia maszynowego i możliwość wykorzystywania go w tworzeniu agenta3. Information Retrieval i boty bazujące na IR4. Rozróżnienie między Sparse i Dense Information Retrieval5. Systemy IR ułatwiające wyszukiwanie zarówno w wariancie rzadkim (np. Elastic Search) jak i gęstym (np. Faiss)6. Odpowiadanie na pytania metodą Reader-Retriever7. Wykorzystanie modeli językowych w odpowiadaniu na pytania, zaawansowane metody konstrukcji promptów8. Możliwości pretrenowanych modeli na przykładzie kolekcji Hugging Face9. Tworzenie botów regulowych, języki AIML oraz ChatScript

15.	Zakładane efekty uczenia się	
	Wiedza	
	Rozumie pojęcie modelu językowego jako rozkładu prawdopodobieństwa oraz jego zastosowanie w systemach dialogowych i odpowiadających na pytania Rozumie, w jaki sposób można teksty reprezentować jako wektory o stałym wymiarze i jakie konsekwencje to ma w tworzeniu agentów konwersacyjnych Zna obecne trendy rozwoju Deep Learningu związane z zastosowaniem dużych modeli językowych (LLM)	K_W01, K_W05 K_W06 K_W05, K_W07
	Umiejętności	
	Umie tworzyć oprogramowanie współpracując w grupie 2-4 osobowej Umie prezentować swoje idee i bronić ich w dyskusji Umie tworzyć programy konwersacyjne i odpowiadające na pytania w paradygmacie regułowym, bazującym na wyszukiwaniu informacji, lub na sieciach neuronowych, jak również stosować kombinacje tych podejść	K_K02, K_U10, K_U04 K_U09 K_U10, K_U11
16.	Literatura obowiązkowa i zalecana <ul style="list-style-type: none"> • Speech and Language Processing (3rd ed. draft) Dan Jurafsky and James H. Martin • Wybrane publikacje dotyczące QA oraz botów z arxiv.com • Natural Language Processing with Transformers: Building Language Applications With Hugging Face, Lewis Tunstall, Leandro von Werra, Thomas Wolf (uzupełniająca) 	
17.	Metody weryfikacji zakładanych efektów kształcenia Prezentacja mini-projektów i projektu końcowego, dyskusje w trakcie prezentacji	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia przedmiotu wymagane jest zdobycie odpowiedniej liczby punktów (określonej w regulaminie), przyznawanej za ukończenie różnych aktywności związanych z mini-projektami i projektem końcowym.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	Praca własna studenta	
	uczestnictwo w zajęciach, konsultowanie i prezentowanie swoich rozwiązań, wspólne dyskusje podczas zajęć	30 godz.
	realizacja projektu końcowego	40 godz.
	realizacja mini-projektów	60 godz.
	samodzielne studiowanie literatury	15 godz.
	Sumarycznie	
	Łączna liczba godzin	175 godz.
	Liczba punktów ECTS	6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt: Deep Learning Project: Deep Learning
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DeepLrn
6.	Rodzaj przedmiotu/modułu Projekt
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty <ul style="list-style-type: none">• Przedmiot przedstawiający tematykę sieci neuronowych. Niezbędne kompetencje <ul style="list-style-type: none">• Umiejętność programowania w języku Python
13.	Cele przedmiotu Celem przedmiotu jest doskonalenie umiejętności programowania metod związanych z głębokim uczeniem (ang. deep learning).
14.	Treści programowe W ramach projektu omówione zostaną zastosowania głębokich sieci neuronowych do przetwarzania tekstu i obrazów. W programie projektu przewiduje się programowanie sieci związanych z klasyfikacją, detekcją obiektów oraz segmentacją obrazów za pomocą takich sieci neuronowych jak: <ul style="list-style-type: none">• głębokie sieci wielowarstwowe (ang. MLP)• głębokie sieci splotowe (ang. deep convolutional neural networks, CNN) Przewiduje się programowanie projektów z wykorzystaniem wielu framework-ów odpowiednich do zadań Machine Learning-owych (ang. MLOPs, Machine Learning operations). W szczególności w projektach należy zapewnić: <ul style="list-style-type: none">• odpowiednią modularność kodu• reprodukowalność eksperymentów poprzez zachowanie wersji kodu, zbiorów danych (dataset registry) i modeli (model registry). Realizacja zadań programistycznych będzie wykorzystwała następujące narzędzia (ang. framework): <ul style="list-style-type: none">• Tensorflow lub PyTorch• keras lub Pytorch lightning do programowania procesów trenowania, odtwarzania eksperymentów, wersjonowania modeli• DVC do wersjonowania plików z danymi lub modelami• Github Actions do utrzymywania ciągłości poprawności działania całości procesu (testy jednostkowe, testy zbiorów danych, testy skuteczności modeli).

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <p>ma pogłębioną wiedzę z zakresu głębokich sieci neuronowych. Dobrze zna rolę i znaczenie ich architektury w modelowaniu danych zna zaawansowane techniki obliczeniowe z wykorzystaniem procesorów CPU i GPU, a także rozumie ich ograniczenia zna narzędzia i technologie stosowane w przemyśle informatycznym do tworzenia oprogramowania i cyklu utrzymywania kodu źródłowego (GitHub), zbiorów danych (ang. dataset registry) i modeli (ang. model registry)</p> <p>Umiejętności</p> <p>potrafi zastosować wiedzę o sieciach neuronowych, do analizowania i rozwiązywania problemów klasyfikacji danych wizyjnych i tekstowych</p> <p>ma umiejętność programowania na poziomie pozwalającym rozwiązywać problemy związane z modelowaniem z użyciem głębokich sieci neuronowych. Posługuje się takimi narzędziami jak framework pytorch i PyTorchLightning posługuje się cyfrową reprezentacją danych: liczb, obrazów i tekstów potrafi implementować programy w zespole</p> <p>Kompetencje społeczne</p> <p>jest świadom możliwości popełniania błędów przez siebie i innych. Wykazuje rozsądny krytycyzm wobec odbieranych treści oraz otrzymywanych wyników, zwłaszcza przy interpretacji wyników uzyskiwanych z użyciem modeli głębokich sieci neuronowych</p>	<p>K_W08</p> <p>K_W07</p> <p>K_W07, K_W10, Inż_W01</p> <p>K_U08, K_U10, K_U13, Inż_U01, Inż_U04</p> <p>K_U03, Inż_U02</p> <p>K_U07, Inż_U04 K_U14, Inż_U03, Inż_U05</p> <p>K_K01</p>
16.	Literatura obowiązkowa i zalecana	
17.	Metody weryfikacji zakładanych efektów kształcenia napisanie programu komputerowego/projekt programistyczny	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia należy zdobyć wymaganą, podaną w regulaminie projektu liczbę punktów za zadania programistyczne i opcjonalną aktywność podczas spotkań (prezentacja i referaty dodatkowych tematów).	
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <p>pracownia</p> <p>Praca własna studenta</p> <p>studiowanie literatury praca nad projektem przygotowanie raportu/prezentacji</p> <p>Sumarycznie</p> <p>Łączna liczba godzin Liczba punktów ECTS</p>	<p>30 godz.</p> <p>40 godz. 80 godz. 10 godz.</p> <p>160 godz. 6</p>

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt dyplomowy Diploma Project
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DP16PrDpl
6.	Rodzaj przedmiotu/modułu Projekt
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów ((jeśli obowiązuje)) —
10.	Semestr nieokreślony
11.	Forma zajęć i liczba godzin pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zaliczenie większości przedmiotów wymaganych do ukończenia studiów I stopnia
13.	Cele przedmiotu Napisanie całości albo fragmentu pracy dyplomowej
14.	Treści programowe Zakres przygotowania pracy dyplomowej obejmuje: <ol style="list-style-type: none"> 1. Sformułowanie problemu będącego tematem pracy dyplomowej 2. Określenie tematu, celu i zakresu pracy dyplomowej 3. Zebranie literatury i zapoznanie się z nią 4. Przystwojenie wiedzy, której dotyczy praca dyplomowa 5. Określenie etapów realizacji pracy dyplomowej 6. Przygotowanie wprowadzenia teoretycznego z zakresu tematu pracy 7. Sporządzenie koncepcji rozwiązania problemu 8. Przyjęcie rozwiązań technicznych, które będą wykorzystane w pracy 9. Implementacja rozwiązania problemu 10. Weryfikacja osiągniętych wyników. 11. Napisanie pracy dyplomowej 12. Prezentacja osiągniętych wyników
15.	Zakładane efekty uczenia się
16.	Literatura obowiązkowa i zalecana <ol style="list-style-type: none"> 1. Informacje ze strony www Instytutu nt rodzajów i zawartości pracy dyplomowej. 2. Literatura wynikająca z tematu i zakresu pracy dyplomowej
17.	Metody weryfikacji zakładanych efektów kształcenia <ul style="list-style-type: none"> • prezentacje dot. realizacji pracy dyplomowej • prezentacja dokumentu praca dyplomowa

18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Aby zaliczyć przedmiot należy:</p> <ul style="list-style-type: none"> • przeprowadzić prezentacje dot. realizacji etapów pracy dyplomowej • przeprowadzić prezentacje dokumentu praca dyplomowa • przedstawić dokument praca dyplomowa 												
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 369 1428 405"> <tr> <td>pracownia</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 481 1428 577"> <tr> <td>Zebranie i analiza literatury</td> <td>10 godz.</td> </tr> <tr> <td>Napisanie pracy dyplomowej</td> <td>20 godz.</td> </tr> <tr> <td>Realizacja etapów pracy dyplomowej</td> <td>60 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 660 1428 723"> <tr> <td>Łączna liczba godzin</td> <td>120 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>4</td> </tr> </table>	pracownia	30 godz.	Zebranie i analiza literatury	10 godz.	Napisanie pracy dyplomowej	20 godz.	Realizacja etapów pracy dyplomowej	60 godz.	Łączna liczba godzin	120 godz.	Liczba punktów ECTS	4
pracownia	30 godz.												
Zebranie i analiza literatury	10 godz.												
Napisanie pracy dyplomowej	20 godz.												
Realizacja etapów pracy dyplomowej	60 godz.												
Łączna liczba godzin	120 godz.												
Liczba punktów ECTS	4												

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt: Kompetytywna Sztuczna Inteligencja Project: Competitive Artificial Intelligence
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DPZCompaAI
6.	Rodzaj przedmiotu/modułu Projekt
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Znajomość tematyki omawianej na przedmiocie Sztuczna Inteligencja (wymagane) oraz Seminarium: Kompetytywna Sztuczna Inteligencja (zalecane), Artificial Intelligence for Games (zalecane).
13.	Cele kształcenia dla przedmiotu Podstawowe cele przedmiotu tu: <ul style="list-style-type: none"> • teoretyczne i praktyczne zaznajomienie studentów z procesem projektowania, programowania i testowania botów • zaznajomienie ze specyfiką gier programistycznych / zawodów AI • rozwinięcie umiejętności pisania algorytmów AI A także nauka współpracy w grupie, wyciągania wniosków z wzajemnych dyskusji, krytycznego myślenia.
14.	Treści programowe <ol style="list-style-type: none"> 1. Zapoznanie się ze specyfiką wybranych zawodów programistycznych AI 2. Organizacja w grupie, wybór tematyki projektu 3. Klasyfikacja gry (pełna informacja, losowość, analiza wielkości drzewa gry) i wiążących się z tym następstw - w szczególności określenie adekwatnych podejść 4. Implementacja wstępnej wersji projektu, testy porównawcze różnych algorytmów AI, dyskusje międzygrupowe 5. Implementacja pełnej wersji bota, optymalizacja wybranych algorytmów 6. Udział w zawodach

15.	Zakładane efekty uczenia się	
	Wiedza	
	zna działanie co najmniej jednego z frameworków wykorzystywanego w zawodach sztucznej inteligencji	K_W08
	rozumie proces powstawania i testowania bota	K_W07
	rozumie specyfikę gier programistycznych, zna klasyfikację gier ze względu na cechy charakterystyczne	K_W10
	zna algorytmy AI i podejścia specyficzne dla rozwiązywania gier	K_W08
	Umiejętności	
	umie zaprojektować, stworzyć oraz przetestować bota grającego w zadaną zasadami konkursu grę	K_U03, K_U08, K_U13, Inż_U02
	umie ocenić treść problemu pod kątem potencjalnie najbardziej skutecznych rozwiązań	K_U07
	umie zaprogramować algorytmy AI przystosowane do różnorodnych zagadnień	K_U03, K_U07, Inż_U03
	umie zaprezentować swój projekt, przedstawić uzasadnienia dla podjętych w czasie jego realizacji decyzji oraz w sposób krytyczny analizować pomysły swoje i innych	K_U12
	potrafi współpracować w grupie i modyfikować wstępne założenia na podstawie otrzymanych informacji zwrotnych	K_U14
	potrafi analizować podejścia innych zespołów oraz przekazywać wartościową informację zwrotną	K_U13, Inż_U03
16.	Literatura obowiązkowa i zalecana	
	Wybrane prace naukowe z dziedziny algorytmów AI, blogi uczestników zawodów, opisy zwycięskich programów, dokumentacja techniczna konkursów.	
17.	Metody weryfikacji zakładanych efektów uczenia się	
	Napisanie i prezentacja programu bota skutecznie grającego w zadaną grę, regularny udział w dyskusjach międzygrupowych, prezentacja finalnego projektu, udział w zawodach jeśli to możliwe (nie wymagane ale zalecane).	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	<ul style="list-style-type: none"> • Uczestnictwo w spotkaniach międzygrupowych, regularne prezentacje postępów prac • Aktywne branie udziału w prezentacjach programów innych grup, konsultowanie pomysłów • Stworzenie i prezentacja gotowego projektu • Napisanie raportu z rozwoju projektu 	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	Praca własna studenta	
	Wybór tematu i zapoznanie się z zasadami wybranych zawodów	10 godz.
	Przygotowanie raportu końcowego	5 godz.
	Prezentacja prac i konsultacje z innymi grupami	20 godz.
	Przygotowanie, implementacja i testy	65 godz.
	Sumarycznie	
	Łączna liczba godzin	130 godz.
	Liczba punktów ECTS	4

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt: Kuźnia rdzeni Project: Coreforge
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-PrKzRdz
6.	Rodzaj przedmiotu/modułu Projekt
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr nieokreślony
11.	Forma zajęć i liczba godzin pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: <ul style="list-style-type: none"> • Logika cyfrowa • Architektury systemów komputerowych • Programowanie układów FPGA (wskazane) • Architektury komputerów (wskazane) Niezbędne kompetencje: <ul style="list-style-type: none"> • Umiejętność projektowania sprzętu przy użyciu co najmniej jednego języka opisu sprzętu. • Teoretyczna znajomość podstaw architektur systemów komputerowych. • Umiejętność programowania niskopoziomowego (wskazane).
13.	Cele przedmiotu Celem projektu jest praktyczna nauka architektur komputerów poprzez ich implementację w formie umożliwiającej syntezę na układy FPGA. Cel ten będzie realizowany poprzez budowę „generatora rdzeniów” — programu, który, wykorzystując zaimplementowane przez studentów moduły, będzie produkował jako wynik układy obliczeniowe o wybranej strukturze i parametrach.
14.	Treści programowe <ul style="list-style-type: none"> • Korzystanie ze standardowych narzędzi programistycznych: np. systemu kontroli wersji, CI (testów automatycznych). • Proces projektowania układu cyfrowego inspirowany procesem tworzenia oprogramowania – ustalenie wymagań, projektowanie, kodowanie, testowanie, code review. • Architektury procesorów out-of-order. • Projektowanie sprzętu przy użyciu języka wysokiego poziomu.

15.	Zakładane efekty uczenia się	
	Wiedza	
	Zna wybrane technologie używane w projektowaniu układów cyfrowych - języki opisu sprzętu, narzędzia do symulacji i syntezy układów, i inne	K_W10
	Zna zasady projektowania układów cyfrowych i potrafi zastosować je w praktyce	K_W10
	Umiejętności	
	Potrafi rozwiązywać problemy napotykane przy projektowaniu układów cyfrowych oraz argumentować użycie wybranych przez siebie technik	K_U08, K_U13
	Potrafi tworzyć, rozwijać i testować moduły układów cyfrowych oraz posługiwać się odpowiednimi do tego narzędziami	K_U03, K_U7
	Umie wykorzystywać narzędzia wspomagające pracę zespołu i jej organizację	K_U14
	Potrafi czytać angielskojęzyczną literaturę dotyczącą rozwiązywanego problemu oraz prezentować zdobytą wiedzę	K_U11
	Kompetencje społeczne	
	Uczestnicząc w spotkaniach projektowych oraz w procesie recenzji kodu, krytycznie analizuje pomysły własne i innych osób	K_K02, K_K06
	Samodzielnie projektuje oraz implementuje rozwiązania zleconych mu zadań, jednocześnie bez oporu komunikuje się z zespołem w przypadku napotkania poważnych trudności	K_K02, K_K05
16.	Literatura obowiązkowa i zalecana <ul style="list-style-type: none"> • Jean-Loup Baer, Microprocessor Architecture - From Simple Pipelines to Chip Multiprocessors. • John Paul Shen, Mikko H. Lipasti, Modern Processor Design - Fundamentals of Superscalar Processors. 	
17.	Metody weryfikacji zakładanych efektów kształcenia Realizacja indywidualnie dobieranych zadań w ramach projektu.	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Uczestnictwo w projekcie wymaga aktywnego udziału w cotygodniowych spotkaniach, opracowywania i złączania zmian, udziału w procesie code review. Zaliczenie przedmiotu wymaga odpowiedniej aktywności w wyżej wymienionych punktach.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	Praca własna studenta	
	analiza kodu, prezentacje, dyskusje	30 godz.
	praca nad projektem	70 godz.
	Sumarycznie	
Łączna liczba godzin	130 godz.	
Liczba punktów ECTS	4	

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt: Machine Learning for Temporal Data Mining Project: Machine Learning for Temporal Data Mining
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DPZMLforTDM
6.	Rodzaj przedmiotu/modułu Projekt
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu wiedza z eksploracji danych i uczenia maszynowego (w zakresie potrzebnym do realizacji projektu) oraz umiejętności techniczne w tym zakresie
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest nabycie/doskonalenie umiejętności pracy zespołowej oraz rozszerzenie wiedzy z algorytmów uczenia maszynowego do eksploracji i modelowania danych temporalnych.
14.	Treści programowe Wybrane algorytmy uczenia maszynowego danych do eksploracji i modelowania danych temporalnych, m.in.: <ul style="list-style-type: none"> • sposoby reprezentacji danych temporalnych i ekstrakcji cech, • algorytmy grupowania i klasyfikacji danych temporalnych, • modelowanie danych temporalnych ze stanami ukrytymi, • zastosowania w systemach wspomagania decyzji.

15.	Zakładane efekty uczenia się	
	Wiedza	
	rozumie wybrane zaawansowane podejścia uczenia maszynowego dla danych temporalnych	K_W08
	rozumie problemy związane z przetwarzaniem danych temporalnych	K_W08, K_W10
	Umiejętności	
	umie formułować zagadnienia praktyczne jako zadania uczenia maszynowego dla danych temporalnych	K_U08, Inż_U03
	umie stosować i modyfikować algorytmy uczenia maszynowego dla danych temporalnych	K_U03, K_U08, Inż_U01, Inż_U02
	umie stosować metody uczenia maszynowego i głębokiego do danych temporalnych	K_U03, K_U08, Inż_U01, Inż_U02
	umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy	K_U12, K_U14, Inż_U03
16.	Literatura obowiązkowa i zalecana	
	wybrane artykuły naukowe z czasopism i konferencji z dziedziny eksploracji danych, inteligencji obliczeniowej i uczenia maszynowego	
17.	Metody weryfikacji zakładanych efektów uczenia się	
	prezentacja, dyskusja	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Zaliczenie wymaga wykonywania przydzielanych indywidualnie zadań projektowych, rozliczania rezultatów i współpracy z grupą.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	Praca własna studenta	
	przygotowywanie się do zajęć (w tym czytanie materiałów dodatkowych)	10 godz.
	przygotowanie prezentacji rezultatów swoich prac	10 godz.
	realizacja zadań projektowych	50 godz.
	Sumarycznie	
	Łączna liczba godzin	100 godz.
	Liczba punktów ECTS	4

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt: Rozwój Systemu Zapisów Project: Development of the Enrollment System
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DP16PrZap
6.	Rodzaj przedmiotu/modułu Projekt
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr nieokreślony
11.	Forma zajęć i liczba godzin pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty <ul style="list-style-type: none"> • Podstawowy warsztat informatyka Niezbędne kompetencje <ul style="list-style-type: none"> • Podstawowa znajomość relacyjnych baz danych (SQL) • Świadomość technologii webowych (HTML, CSS, Javascript/TypeScript) • Podstawy Pythona
13.	Cele kształcenia dla przedmiotu Rozróżniamy dwa podstawowe cele przedmiotu. Z perspektywy prowadzącego celem jest rozwijanie Systemu Zapisów w naszym instytucie i wykorzystanie do tego celu zdolności oraz zapału studentów. Studenci jednocześnie (drugi cel) uczą się uczestnictwa w dużym projekcie programistycznym, do którego przychodzą, gdy projekt jest już dość rozwinięty. Muszą wykonać w nim pewne zadania, zmagając się jednocześnie z niedoskonałościami tego projektu. Jest to scenariusz bardziej odpowiadający realiom pracy zawodowej, niż na większości kursów, gdzie studenci tworzą projekt „od zera”.
14.	Treści programowe <ul style="list-style-type: none"> • Korzystanie ze standardowych narzędzi programistycznych: np. systemu kontroli wersji, CI (testów automatycznych). • Typowy proces tworzenia oprogramowania — zadania, pomysły, projektowanie, kodowanie, weryfikowanie (code review, testy), dołączanie. • Standardowe problemy i rozwiązania w oprogramowaniu webowym — bezpieczeństwo, radzenie sobie z równoległością, wątki, wydajność. • Modne technologie webowe: Django, Vue.

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <p>Zna wybrane technologie używane w aplikacjach webowych (serwerowych). Rozumie zasady działania relacyjnych baz danych i umie efektywnie z nich korzystać.</p> <p>Umiejętności</p> <p>Potrafi rozwijać, uruchamiać i testować oprogramowanie w zadanej technologii. Projektuje rozwiązanie dla postawionego zadania i prezentuje to rozwiązanie. Potrafi ocenić oraz uargumentować zasadność stosowania różnych narzędzi i technik.</p> <p>Potrafi zidentyfikować istotny dla siebie fragment kodu w dużym systemie informatycznym i przeanalizować jego działanie.</p> <p>Umie używać popularnych narzędzi do pracy zespołu programistycznego (takich jak Slack, Github).</p> <p>Umie zrobić rzeczowe „code-review” i nie przyjmuje postawy defensywnej, gdy ktoś zrobi „code-review” jemu.</p> <p>Umie korzystać z dokumentacji oraz znajdować w Internecie rozwiązania przy napotkanych przeszkodach technicznych.</p>	<p>K_W08, K_W10 K_W07, K_W08</p> <p>K_U03, Inż_U05 K_U03, K_U08, Inż_U05</p> <p>K_U09, Inż_U01</p> <p>K_U07, Inż_U02, Inż_U04</p> <p>K_U07, Inż_U02, Inż_U03, Inż_U04</p> <p>K_U07, Inż_U02, Inż_U04</p>								
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • Dokumentacja Django (https://docs.djangoproject.com/en/). • Dokumentacja Vue (https://vuejs.org/v2/guide/). <p>Oczywiście dokumentacji nie trzeba czytać w całości.</p>									
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>Realizacja zadań w Systemie Zapisów.</p>									
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Projekt programistyczny, cotygodniowa ewaluacja pracy na spotkaniach.</p>									
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 1200 1428 1234"> <tr> <td>pracownia</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 1317 1428 1350"> <tr> <td>Praca nad projektem</td> <td>70 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 1435 1428 1496"> <tr> <td>Łączna liczba godzin</td> <td>100 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>4</td> </tr> </table>	pracownia	30 godz.	Praca nad projektem	70 godz.	Łączna liczba godzin	100 godz.	Liczba punktów ECTS	4	
pracownia	30 godz.									
Praca nad projektem	70 godz.									
Łączna liczba godzin	100 godz.									
Liczba punktów ECTS	4									

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt: silnik szachowy Project: Chess Engine
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DPSzachy
6.	Rodzaj przedmiotu/modułu Projekt
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: <ul style="list-style-type: none">• Sztuczna inteligencja Niezbędne kompetencje: <ul style="list-style-type: none">• znajomość podstawowych algorytmów przeszukiwania w grach, i.e., alfa-beta obcięcia oraz Monte Carlo Tree Search
13.	Cele przedmiotu Podstawowym celem przedmiotu jest zaznajomienie studentów z nowożytnymi technikami konstrukcji programów grających w szachy. Tego rodzaju silniki wykorzystują piękny blend klasycznych metod przeszukiwań drzewa gry, jak alfa-beta obcięcia czy MCTS, ale korzystają z funkcji ewaluacji, które pochodzą ze świata deep learningu. Szczególnie ciekawe w tym kontekście jest, że podejścia te, a zwłaszcza podejście oparte na MCTS, wykorzystują bardzo generyczną metodę samogrania, czy też self-playu, która to jest znacznie bardziej generyczna i daje się również wykorzystywać w wielu innych problemach, jak na przykład efektywne mnożenie macierzy.
14.	Treści programowe <ol style="list-style-type: none">1. Klasyczne algorytmy przeszukiwań, alfa-beta obcięcia i MCTS2. Sieć neuronowa, która uczy się poprawnej, choć rzecz jasna przybliżonej, ewaluacji stanów3. Implementacja klasycznej metody przeszukiwania wzbogaconą o deep learningową funkcję ewaluacji stanów4. Poprawianie precyzji sieci neuronowej za pomocą self-playu

15.	Zakładane efekty uczenia się	
	Wiedza	
	wie jak działają klasyczne metody przeszukiwań — alfa-beta i MCTS zaawansowana znajomość działania self-play zna narzędzia stosowane w przemyśle informatycznym do tworzenia oprogramowania i utrzymywania kodu źródłowego	K_W07 K_W07 K_W10, Inż_W01
	Umiejętności	
	potrafi zaimplementować silnik gry w szachy	K_U03, K_U08, K_U13, Inż_U01, Inż_U02
	potrafi stosować sieci neuronowe do reprezentacji strategii gry w szachy	K_U07, Inż_U01, Inż_U04
	umie tworzyć oprogramowanie współpracując w grupie	K_U14, Inż_U03, Inż_U05
16.	Literatura obowiązkowa i zalecana <ul style="list-style-type: none"> • Chess Algorithms, Noah Caplinger • Neural Networks for Chess, Dominik Klein 	
17.	Metody weryfikacji zakładanych efektów kształcenia prezentacja projektu	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Napisanie poprawnej i efektywnej części silnika — metody przeszukiwania bądź sieci neuronowej.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	Praca własna studenta	
	praca nad ustalonym modułem silnika	65 godz.
	Sumarycznie	
	Łączna liczba godzin	95 godz.
	Liczba punktów ECTS	4

8 Pozostałe zajęcia

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Etyka dla myślących Ethics for the thinking
2.	Dyscyplina filozofia
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Katedra Logiki i Metodologii Nauk
5.	Kod przedmiotu/modułu 28-INF-S-EtkdMsl
6.	Rodzaj przedmiotu/modułu Humanistyczno-społeczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele kształcenia dla przedmiotu <ul style="list-style-type: none"> • Zapoznanie studenta z najważniejszymi koncepcjami etyki normatywnej; • Wdrożenie studenta do posługiwania się pojęciami etycznymi niezbędnymi do zbudowania świadomego światopoglądu moralnego; • WYROBIENIE umiejętności krytycznej lektury tekstu filozoficznego; • WYROBIENIE umiejętności racjonalnej argumentacji na rzecz przyjętej hipotezy; • DOSKONALENIE umiejętności pisemnego wypowiedzania się na tematy o wysokim stopniu abstrakcyjności.

14.	<p>Treści programowe</p> <p>W trakcie wykładów zostaną omówione najważniejsze szkoły i koncepcje etyki filozoficznej ze szczególnym podkreśleniem wybranego pojęcia moralnego.</p> <p>Ćwiczenia będą polegały na krytycznym dyskutowaniu koncepcji omawianych na wykładach.</p> <ol style="list-style-type: none"> 1. Wprowadzenie do etyki jako dziedziny filozoficznej. Etyka a moralność, prawo i obyczaj. Etyka opisowa, normatywna i metaetyka. Pojęcie moralności. 2. Początki refleksji moralnej: Homer i Sokrates. Pojęcie honoru. 3. Platon – sprawiedliwy człowiek, sprawiedliwe państwo. Pojęcie sprawiedliwości. 4. Arystoteles – etyka złotego środka. Pojęcie cnoty. 5. Filozofowie w poszukiwaniu szczęścia: stoicyzm, epikureizm, cynizm, hedonizm. Pojęcie szczęścia. 6. Wiara i rozum – św. Augustyn i św. Tomasz z Akwinu. Pojęcie sumienia. 7. Uczucia moralne – Hume. Pojęcie tolerancji. 8. Etyka umowy społecznej. Pojęcie godności. 9. Imperatyw kategoryczny Kanta. Pojęcie obowiązku. 10. Zasada użyteczności. Pojęcie wolności. 11. Egzystencjalizm. Nietzsche i Scheler. Pojęcie sensu życia. 12. Szkoła lwowsko-warszawska. Kotarbiński i Czeżowski. Pojęcie dylematu moralnego. 13. Elzenberg i etyka wartości. Pojęcie wartości. 14. Sens i znaczenie ocen i norm moralnych. Główne kierunki metaetyki. Pojęcie normy i oceny moralnej. 15. Klasyfikacja stanowisk etycznych. Podsumowanie. 										
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 882 1426 1010"> <tr> <td data-bbox="225 882 1166 913">Student ma podstawową wiedzę z zakresu etyki filozoficznej</td> <td data-bbox="1174 882 1426 913">K_W09</td> </tr> <tr> <td data-bbox="225 913 1166 945">Student zna podstawową terminologię etyczną</td> <td data-bbox="1174 913 1426 945">K_W09</td> </tr> <tr> <td data-bbox="225 945 1166 1010">Student zna i rozumie rolę etyki w kształtowaniu świadomych postaw moralnych.</td> <td data-bbox="1174 945 1426 1010">K_W09</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 1095 1426 1285"> <tr> <td data-bbox="225 1095 1166 1285"> <p>Student potrafi przeczytać, przedyskutować i skategoryzować dany tekst etyczny</p> <p>Student potrafi poprawnie rozpoznać i zastosować wybraną terminologię etyczną</p> <p>Student potrafi racjonalnie wypowiedzieć się słownie oraz pisemnie na temat wybranych zagadnień moralnych.</p> </td> <td data-bbox="1174 1095 1426 1285"></td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 1370 1426 1561"> <tr> <td data-bbox="225 1370 1166 1561"> <p>Student jest zdolny do samodzielnego budowania systemu własnych poglądów i formułowania krytycznych ocen</p> <p>Student dzięki posiadanej wiedzy i umiejętnościom etycznym rozumie potrzebę ciągłego rozwoju osobistego</p> <p>Student ma świadomość znaczenia europejskiego dziedzictwa etycznego i potrafi je docenić.</p> </td> <td data-bbox="1174 1370 1426 1561"></td> </tr> </table>	Student ma podstawową wiedzę z zakresu etyki filozoficznej	K_W09	Student zna podstawową terminologię etyczną	K_W09	Student zna i rozumie rolę etyki w kształtowaniu świadomych postaw moralnych.	K_W09	<p>Student potrafi przeczytać, przedyskutować i skategoryzować dany tekst etyczny</p> <p>Student potrafi poprawnie rozpoznać i zastosować wybraną terminologię etyczną</p> <p>Student potrafi racjonalnie wypowiedzieć się słownie oraz pisemnie na temat wybranych zagadnień moralnych.</p>		<p>Student jest zdolny do samodzielnego budowania systemu własnych poglądów i formułowania krytycznych ocen</p> <p>Student dzięki posiadanej wiedzy i umiejętnościom etycznym rozumie potrzebę ciągłego rozwoju osobistego</p> <p>Student ma świadomość znaczenia europejskiego dziedzictwa etycznego i potrafi je docenić.</p>	
Student ma podstawową wiedzę z zakresu etyki filozoficznej	K_W09										
Student zna podstawową terminologię etyczną	K_W09										
Student zna i rozumie rolę etyki w kształtowaniu świadomych postaw moralnych.	K_W09										
<p>Student potrafi przeczytać, przedyskutować i skategoryzować dany tekst etyczny</p> <p>Student potrafi poprawnie rozpoznać i zastosować wybraną terminologię etyczną</p> <p>Student potrafi racjonalnie wypowiedzieć się słownie oraz pisemnie na temat wybranych zagadnień moralnych.</p>											
<p>Student jest zdolny do samodzielnego budowania systemu własnych poglądów i formułowania krytycznych ocen</p> <p>Student dzięki posiadanej wiedzy i umiejętnościom etycznym rozumie potrzebę ciągłego rozwoju osobistego</p> <p>Student ma świadomość znaczenia europejskiego dziedzictwa etycznego i potrafi je docenić.</p>											

16.	<p>Literatura obowiązkowa i zalecana</p> <p>Literatura podstawowa:</p> <ul style="list-style-type: none"> • M. Środa, Etyka dla myślących. Podręcznik dla szkół ponadgimnazjalnych, Warszawa 2016. • Platon, Obrona Sokratesa (różne wydania). • Platon, Państwo, ks. 4 i 7. • Arystoteles, Etyka nikomachejska, ks. 2-3. • M. Aureliusz, Rozmyślenia, ks. 2-5 • św. Augustyn, Wyznania ks. 7,5, Tomasz z Akwinu, Summa, kw. 80, 83 i 90. • D. Hume, Badania dotyczące zasad moralności, cz. 1-5. • T. Hobbes, Lewiatan, rozdz. 10-21, J. Rawls, Teoria sprawiedliwości, par. 1-4, 11-17, 62-63. • I. Kant, Uzasadnienie metafizyki moralności, rozdz. 1-2. • J. S. Mill, Utylitaryzm. • F. Nietzsche, Z genealogii moralności, cz. I. • H. Elzenberg, Wartość i powinność, w: O wartościach, normach i problemach moralnych, (red.) M. Środa, Warszawa 1994. • T. Czeżowski, Aksjologiczne i deontyczne normy moralne, w: O wartościach, normach i problemach moralnych, (red.) M. Środa, Warszawa 1994. • A. Pap, Teorie metaetyczne, w: Metaetyka, (red.) I. Lazari-Pawłowska, Warszawa 1975. <p>Literatura dodatkowa:</p> <ul style="list-style-type: none"> • A. MacIntyre, Krótka historia etyki, Warszawa 2013. • P. Vardy, P. Grosch, Etyka, Poglądy i problemy, Poznań 2010. 												
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <ul style="list-style-type: none"> • egzamin ustny, • przygotowywanie wypowiedzi pisemnych. 												
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <ul style="list-style-type: none"> • Wykład: egzamin ustny (trzy pytania na podstawie przygotowanych uprzednio przez studenta konspektów lektur źródłowych). Obecność. • Ćwiczenia: obecność oraz przedstawianie pięciu pisemnych wypowiedzi na zadane do zajęć pytania. 												
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 1249 1422 1317"> <tr> <td>wykład</td> <td>30 godz.</td> </tr> <tr> <td>ćwiczenia</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 1391 1422 1458"> <tr> <td>przygotowanie do zajęć (pisanie wypowiedzi pisemnych)</td> <td>30 godz.</td> </tr> <tr> <td>przygotowanie do egzaminu (czytanie wskazanej literatury)</td> <td>30 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 1541 1422 1601"> <tr> <td>Łączna liczba godzin</td> <td>120 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>5</td> </tr> </table>	wykład	30 godz.	ćwiczenia	30 godz.	przygotowanie do zajęć (pisanie wypowiedzi pisemnych)	30 godz.	przygotowanie do egzaminu (czytanie wskazanej literatury)	30 godz.	Łączna liczba godzin	120 godz.	Liczba punktów ECTS	5
wykład	30 godz.												
ćwiczenia	30 godz.												
przygotowanie do zajęć (pisanie wypowiedzi pisemnych)	30 godz.												
przygotowanie do egzaminu (czytanie wskazanej literatury)	30 godz.												
Łączna liczba godzin	120 godz.												
Liczba punktów ECTS	5												

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Etyka dla myślących (w) Ethics for the thinking
2.	Dyscyplina filozofia
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Katedra Logiki i Metodologii Nauk
5.	Kod przedmiotu/modułu 28-INF-S-EtkdMsl
6.	Rodzaj przedmiotu/modułu Humanistyczno-społeczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele kształcenia dla przedmiotu <ul style="list-style-type: none"> • Zapoznanie studenta z najważniejszymi koncepcjami etyki normatywnej; • Wdrożenie studenta do posługiwania się pojęciami etycznymi potrzebnymi do zbudowania świadomego światopoglądu moralnego; • WYROBIENIE UMIEJĘTNOŚCI KRYTYCZNEJ LEKTURY TEKSTU FILOZOFICZNEGO; • WYROBIENIE UMIEJĘTNOŚCI RACJONALNEJ ARGUMENTACJI NA RZECZ PRZYJĘTEJ HIPOTEZY; • DOSKONALENIE UMIEJĘTNOŚCI PISEMNEGO WYPOWIADANIA SIĘ NA TEMATY O WYSOKIM STOPNIU ABSTRAKCYJNOŚCI.

14.	<p>Treści programowe</p> <p>W trakcie wykładów zostaną omówione najważniejsze szkoły i koncepcje etyki filozoficznej ze szczególnym podkreśleniem wybranego pojęcia moralnego.</p> <p>Ćwiczenia będą polegały na krytycznym dyskutowaniu koncepcji omawianych na wykładach.</p> <ol style="list-style-type: none"> 1. Wprowadzenie do etyki jako dziedziny filozoficznej. Etyka a moralność, prawo i obyczaj. Etyka opisowa, normatywna i metaetyka. Pojęcie moralności. 2. Początki refleksji moralnej: Homer i Sokrates. Pojęcie honoru. 3. Platon – sprawiedliwy człowiek, sprawiedliwe państwo. Pojęcie sprawiedliwości. 4. Arystoteles – etyka złotego środka. Pojęcie cnoty. 5. Filozofowie w poszukiwaniu szczęścia: stoicyzm, epikureizm, cynizm, hedonizm. Pojęcie szczęścia. 6. Wiara i rozum – św. Augustyn i św. Tomasz z Akwinu. Pojęcie sumienia. 7. Uczucia moralne – Hume. Pojęcie tolerancji. 8. Etyka umowy społecznej. Pojęcie godności. 9. Imperatyw kategoryczny Kanta. Pojęcie obowiązku. 10. Zasada użyteczności. Pojęcie wolności. 11. Egzystencjalizm. Nietzsche i Scheler. Pojęcie sensu życia. 12. Szkoła lwowsko-warszawska. Kotarbiński i Czeżowski. Pojęcie dylematu moralnego. 13. Elzenberg i etyka wartości. Pojęcie wartości. 14. Sens i znaczenie ocen i norm moralnych. Główne kierunki metaetyki. Pojęcie normy i oceny moralnej. 15. Klasyfikacja stanowisk etycznych. Podsumowanie. 																		
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 882 1426 1010"> <tr> <td data-bbox="225 882 1166 913">Student ma podstawową wiedzę z zakresu etyki filozoficznej</td> <td data-bbox="1174 882 1426 913">K_W09</td> </tr> <tr> <td data-bbox="225 913 1166 945">Student zna podstawową terminologię etyczną</td> <td data-bbox="1174 913 1426 945">K_W09</td> </tr> <tr> <td data-bbox="225 945 1166 1010">Student zna i rozumie rolę etyki w kształtowaniu świadomych postaw moralnych.</td> <td data-bbox="1174 945 1426 1010">K_W09</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 1095 1426 1285"> <tr> <td data-bbox="225 1095 1166 1155">Student potrafi przeczytać, przedyskutować i skategoryzować dany tekst etyczny</td> <td data-bbox="1174 1095 1426 1285"></td> </tr> <tr> <td data-bbox="225 1155 1166 1216">Student potrafi poprawnie rozpoznać i zastosować wybraną terminologię etyczną</td> <td data-bbox="1174 1155 1426 1285"></td> </tr> <tr> <td data-bbox="225 1216 1166 1285">Student potrafi racjonalnie wypowiedzieć się słownie oraz pisemnie na temat wybranych zagadnień moralnych.</td> <td data-bbox="1174 1216 1426 1285"></td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 1370 1426 1561"> <tr> <td data-bbox="225 1370 1166 1431">Student jest zdolny do samodzielnego budowania systemu własnych poglądów i formułowania krytycznych ocen</td> <td data-bbox="1174 1370 1426 1561"></td> </tr> <tr> <td data-bbox="225 1431 1166 1491">Student dzięki posiadanej wiedzy i umiejętnościom etycznym rozumie potrzebę ciągłego rozwoju osobistego</td> <td data-bbox="1174 1431 1426 1561"></td> </tr> <tr> <td data-bbox="225 1491 1166 1561">Student ma świadomość znaczenia europejskiego dziedzictwa etycznego i potrafi je docenić.</td> <td data-bbox="1174 1491 1426 1561"></td> </tr> </table>	Student ma podstawową wiedzę z zakresu etyki filozoficznej	K_W09	Student zna podstawową terminologię etyczną	K_W09	Student zna i rozumie rolę etyki w kształtowaniu świadomych postaw moralnych.	K_W09	Student potrafi przeczytać, przedyskutować i skategoryzować dany tekst etyczny		Student potrafi poprawnie rozpoznać i zastosować wybraną terminologię etyczną		Student potrafi racjonalnie wypowiedzieć się słownie oraz pisemnie na temat wybranych zagadnień moralnych.		Student jest zdolny do samodzielnego budowania systemu własnych poglądów i formułowania krytycznych ocen		Student dzięki posiadanej wiedzy i umiejętnościom etycznym rozumie potrzebę ciągłego rozwoju osobistego		Student ma świadomość znaczenia europejskiego dziedzictwa etycznego i potrafi je docenić.	
Student ma podstawową wiedzę z zakresu etyki filozoficznej	K_W09																		
Student zna podstawową terminologię etyczną	K_W09																		
Student zna i rozumie rolę etyki w kształtowaniu świadomych postaw moralnych.	K_W09																		
Student potrafi przeczytać, przedyskutować i skategoryzować dany tekst etyczny																			
Student potrafi poprawnie rozpoznać i zastosować wybraną terminologię etyczną																			
Student potrafi racjonalnie wypowiedzieć się słownie oraz pisemnie na temat wybranych zagadnień moralnych.																			
Student jest zdolny do samodzielnego budowania systemu własnych poglądów i formułowania krytycznych ocen																			
Student dzięki posiadanej wiedzy i umiejętnościom etycznym rozumie potrzebę ciągłego rozwoju osobistego																			
Student ma świadomość znaczenia europejskiego dziedzictwa etycznego i potrafi je docenić.																			

16.	<p>Literatura obowiązkowa i zalecana</p> <p>Literatura podstawowa:</p> <ul style="list-style-type: none"> • M. Środa, Etyka dla myślących. Podręcznik dla szkół ponadgimnazjalnych, Warszawa 2016. • Platon, Obrona Sokratesa (różne wydania). • Platon, Państwo, ks. 4 i 7. • Arystoteles, Etyka nikomachejska, ks. 2-3. • M. Aureliusz, Rozmyślenia, ks. 2-5 • św. Augustyn, Wyznania ks. 7,5, Tomasz z Akwinu, Summa, kw. 80, 83 i 90. • D. Hume, Badania dotyczące zasad moralności, cz. 1-5. • T. Hobbes, Lewiatan, rozdz. 10-21, J. Rawls, Teoria sprawiedliwości, par. 1-4, 11-17, 62-63. • I. Kant, Uzasadnienie metafizyki moralności, rozdz. 1-2. • J. S. Mill, Utylitaryzm. • F. Nietzsche, Z genealogii moralności, cz. I. • H. Elzenberg, Wartość i powinność, w: O wartościach, normach i problemach moralnych, (red.) M. Środa, Warszawa 1994. • T. Czeżowski, Aksjologiczne i deontyczne normy moralne, w: O wartościach, normach i problemach moralnych, (red.) M. Środa, Warszawa 1994. • A. Pap, Teorie metaetyczne, w: Metaetyka, (red.) I. Lazari-Pawłowska, Warszawa 1975. <p>Literatura dodatkowa:</p> <ul style="list-style-type: none"> • A. MacIntyre, Krótka historia etyki, Warszawa 2013. • P. Vardy, P. Grosch, Etyka, Poglądy i problemy, Poznań 2010. 								
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <ul style="list-style-type: none"> • egzamin ustny 								
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <ul style="list-style-type: none"> • Wykład: egzamin ustny (trzy pytania na podstawie przygotowanych uprzednio przez studenta konspektów lektur źródłowych). Obecność. 								
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="223 1164 1428 1205"> <tr> <td>wykład</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="223 1276 1428 1317"> <tr> <td>przygotowanie do egzaminu (czytanie wskazanej literatury)</td> <td>40 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="223 1400 1428 1467"> <tr> <td>Łączna liczba godzin</td> <td>70 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>3</td> </tr> </table>	wykład	30 godz.	przygotowanie do egzaminu (czytanie wskazanej literatury)	40 godz.	Łączna liczba godzin	70 godz.	Liczba punktów ECTS	3
wykład	30 godz.								
przygotowanie do egzaminu (czytanie wskazanej literatury)	40 godz.								
Łączna liczba godzin	70 godz.								
Liczba punktów ECTS	3								

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Historia filozofii History of Philosophy
2.	Dyscyplina filozofia
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Katedra Logiki i Metodologii Nauk
5.	Kod przedmiotu/modułu 28-INF-S-HistFiloz
6.	Rodzaj przedmiotu/modułu Humanistyczno-społeczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest kształtowanie umiejętności samodzielnego myślenia na kanwie wiedzy zdobytej podczas wykładów i umiejętności nabytych na ćwiczeniach, kształtowanie wrażliwości na różnego rodzaju argumentację obecną w całości antroposfery, rozwijanie umiejętności krytycznego myślenia, praktyczne przygotowanie do wstępnej, samodzielnej lektury tekstów z zakresu historii filozofii.

14.	<p>Treści programowe</p> <ol style="list-style-type: none"> 1. Charakterystyka filozofii i jej subdyscyplin, geneza nazwy. 2. Presokratycy, sofisci, Sokrates – poszukiwanie przedmiotu filozofii. 3. Platon i Arystoteles – systemy świata starożytnego. 4. Stoicy i epikurejczycy – charakterystyka myśli hellenistycznej. 5. Wczesne średniowiecze i św. Augustyn. 6. Scholastyka i św. Tomasz. 7. Renesans i początek nowożytności: Machiavelli, Pico della Mirandola, Hobbes. 8. Kartezjusz – „ojciec filozofii nowożytnej”. 9. Leibniz i Spinoza: wokół substancji; lub Locke, Berkeley, Hume: wokół percepcji. 10. Kant - „przewrót kopernikański” i Schopenhauer - „świat jako wola i wyobrażenie” 11. Hegel: historia i dialektyka oraz krytycy jego filozofii zapośredniczeni w niej: Kierkegaard, Marks, Feuerbach. 12. Pozytywizm i filozofia nauki – Comte, Mach, Koło Wiedeńskie, Wittgenstein, Kuhn, Popper. 13. Różne oblicza współczesności: Fenomenologia i pragmatyzm; filozofia chrześcijańska i marksizm; poststrukturalizm i personalizm; postmodernizm i dekonstrukcjonizm. 14. Wokół egzystencjalizm: Szestow, Heidegger, Jaspers, Sartre’a; lub filozofia życia: Nietzsche, Dilthey, Bergson. 15. Cum grano salis: wokół „partii matematycznej” w filozofii: o pitagorejczykach, Platonie, Kartezjuszu, Spinozie, Kancie, Whiteheadzie, Russellu... pod kątem ich przekonania o możliwość wykorzystania rozwiązań matematycznych do rozstrzygnięć ontologicznych i metafizycznych, jak również etycznych i politycznych. 								
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="209 913 1437 1070"> <tr> <td data-bbox="209 913 1166 1003">Posiada podstawową wiedzę o specyfice i roli filozofii oraz podstawowych zmianach w obrębie poruszanej przez nią problematyki od starożytności do współczesności</td> <td data-bbox="1171 913 1437 1003">K_W09</td> </tr> <tr> <td data-bbox="209 1003 1166 1070">Charakteryzuje różne - omawiane na ćwiczeniach i wykładzie - koncepcje filozoficzne; wymienia najbardziej charakterystyczne dla nich założenia</td> <td data-bbox="1171 1003 1437 1070">K_W09</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="209 1160 1437 1256"> <tr> <td data-bbox="209 1160 1166 1256">Potrafi analizować tekst, a także interpretować problemy w nim zawarte w szerszym tle odniesień do kultury współczesnej i własnej wrażliwości Stosuje poprawnie podstawowe pojęcia z zakresu filozofii</td> <td data-bbox="1171 1160 1437 1256"></td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="209 1339 1437 1431"> <tr> <td data-bbox="209 1339 1166 1431">Przejawia otwartość i wrażliwość na różnego rodzaju argumentacje i dostrzega ograniczenia różnego typu argumentacji, zwłaszcza tych które uznaje za optymalne</td> <td data-bbox="1171 1339 1437 1431"></td> </tr> </table>	Posiada podstawową wiedzę o specyfice i roli filozofii oraz podstawowych zmianach w obrębie poruszanej przez nią problematyki od starożytności do współczesności	K_W09	Charakteryzuje różne - omawiane na ćwiczeniach i wykładzie - koncepcje filozoficzne; wymienia najbardziej charakterystyczne dla nich założenia	K_W09	Potrafi analizować tekst, a także interpretować problemy w nim zawarte w szerszym tle odniesień do kultury współczesnej i własnej wrażliwości Stosuje poprawnie podstawowe pojęcia z zakresu filozofii		Przejawia otwartość i wrażliwość na różnego rodzaju argumentacje i dostrzega ograniczenia różnego typu argumentacji, zwłaszcza tych które uznaje za optymalne	
Posiada podstawową wiedzę o specyfice i roli filozofii oraz podstawowych zmianach w obrębie poruszanej przez nią problematyki od starożytności do współczesności	K_W09								
Charakteryzuje różne - omawiane na ćwiczeniach i wykładzie - koncepcje filozoficzne; wymienia najbardziej charakterystyczne dla nich założenia	K_W09								
Potrafi analizować tekst, a także interpretować problemy w nim zawarte w szerszym tle odniesień do kultury współczesnej i własnej wrażliwości Stosuje poprawnie podstawowe pojęcia z zakresu filozofii									
Przejawia otwartość i wrażliwość na różnego rodzaju argumentacje i dostrzega ograniczenia różnego typu argumentacji, zwłaszcza tych które uznaje za optymalne									

16.	<p>Literatura obowiązkowa i zalecana</p> <p>Literatura obowiązkowa do ćwiczeń:</p> <ol style="list-style-type: none"> 1. Obraz filozofii i filozofa: „Obrona Sokratesa” Platona 2. Wokół filozofii starożytnego świata: Wybrane fragmenty z „Państwa” Platona (rozdział 7) i „Metafizyki” Arystotelesa z ksiąg II i IV 3. Wokół filozofii średniowiecznego świata: Wybrane fragmenty z Augustyna „Soliloquwów” (rozdział 2,6) i Tomasza „O królowaniu” (rozdział 13-15) 4. Dylematy nowożytności: Wybrany - przez studentów - jeden blok: 1.) „Rozprawy o metodzie” Descartes’a i „Etyki” Spinozy lub 2.) Pascala „Myśli” lub 3.) Wolter „Kandyd, czyli optymizm” lub 4.) Locke „List o tolerancji” 5. Systemy nowożytnego świata: Wybrany - przez studentów - jeden blok: 1.) fragmenty z „Monadologii” Leibniza lub 2.) fragmenty z „Prolegomeny” Kanta lub 3.) fragmenty z „Nauki logiki” Hegla 6. Różne oblicza współczesności: Wybrany - przez studentów - jeden blok 1.) Kuhn „Struktura rewolucji naukowych” (roz. V „Priorytet paradygmatów” lub 2.) Fromm „Ucieczka od wolności” (roz. IV „Dwa aspekty wolności człowieka współczesnego”) lub 3.) J. P. Sartre „Byt i Nicość” (Wprowadzenie: W poszukiwaniu bytu) lub 4.) Th. W. Adorno „Teoria estetyczna” („Wczesne wprowadzenie”) 7. O filozofii, nauce, matematyce i życiu na wesoło: H. Steinhaus „Słownik racjonalny” Literatura uzupełniająca (do wykładu) wybrany podręcznik: <ul style="list-style-type: none"> • W. Tatarkiewicz „Historia filozofii” t. 1-3 • S. Swieżawski „Dzieje europejskiej filozofii klasycznej” • B. Russell „Dzieje filozofii zachodu” 																
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>egzamin ustny, dyskusja, praca pisemna</p>																
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Ćwiczenia: obecność, aktywność (ocenie ciągle), na ocenę bdb. - miniesej 2-3 strony.</p> <p>Wykład: egzamin ustny weryfikujący opanowanie wiedzy i umiejętności.</p>																
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 1162 1426 1229"> <tr> <td>wykład</td> <td>30 godz.</td> </tr> <tr> <td>ćwiczenia</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 1308 1426 1435"> <tr> <td>przygotowanie do ćwiczeń</td> <td>30 godz.</td> </tr> <tr> <td>studiowanie tematyki wykładów i literatury</td> <td>30 godz.</td> </tr> <tr> <td>przygotowanie do egzaminu</td> <td>10 godz.</td> </tr> <tr> <td>przygotowanie raportu/prezentacji</td> <td>10 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 1514 1426 1581"> <tr> <td>Łączna liczba godzin</td> <td>140 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>5</td> </tr> </table>	wykład	30 godz.	ćwiczenia	30 godz.	przygotowanie do ćwiczeń	30 godz.	studiowanie tematyki wykładów i literatury	30 godz.	przygotowanie do egzaminu	10 godz.	przygotowanie raportu/prezentacji	10 godz.	Łączna liczba godzin	140 godz.	Liczba punktów ECTS	5
wykład	30 godz.																
ćwiczenia	30 godz.																
przygotowanie do ćwiczeń	30 godz.																
studiowanie tematyki wykładów i literatury	30 godz.																
przygotowanie do egzaminu	10 godz.																
przygotowanie raportu/prezentacji	10 godz.																
Łączna liczba godzin	140 godz.																
Liczba punktów ECTS	5																

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Historia filozofii (w) History of Philosophy
2.	Dyscyplina filozofia
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Katedra Logiki i Metodologii Nauk
5.	Kod przedmiotu/modułu 28-INF-S-HistFiloz
6.	Rodzaj przedmiotu/modułu Humanistyczno-społeczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest kształtowanie umiejętności samodzielnego myślenia na kanwie wiedzy zdobytej podczas wykładów i umiejętności nabytych na ćwiczeniach, kształtowanie wrażliwości na różnego rodzaju argumentację obecną w całości antroposfery, rozwijanie umiejętności krytycznego myślenia, praktyczne przygotowanie do wstępnej, samodzielnej lektury tekstów z zakresu historii filozofii.

14.	<p>Treści programowe</p> <ol style="list-style-type: none"> 1. Charakterystyka filozofii i jej subdyscyplin, geneza nazwy. 2. Presokratycy, sofisci, Sokrates – poszukiwanie przedmiotu filozofii. 3. Platon i Arystoteles – systemy świata starożytnego. 4. Stoicy i epikurejczycy – charakterystyka myśli hellenistycznej. 5. Wczesne średniowiecze i św. Augustyn. 6. Scholastyka i św. Tomasz. 7. Renesans i początek nowożytności: Machiavelli, Pico della Mirandola, Hobbes. 8. Kartezjusz – „ojciec filozofii nowożytnej”. 9. Leibniz i Spinoza: wokół substancji; lub Locke, Berkeley, Hume: wokół percepcji. 10. Kant - „przewrót kopernikański” i Schopenhauer - „świat jako wola i wyobrażenie” 11. Hegel: historia i dialektyka oraz krytycy jego filozofii zapośredniczeni w niej: Kierkegaard, Marks, Feuerbach. 12. Pozytywizm i filozofia nauki – Comte, Mach, Koło Wiedeńskie, Wittgenstein, Kuhn, Popper. 13. Różne oblicza współczesności: Fenomenologia i pragmatyzm; filozofia chrześcijańska i marksizm; poststrukturalizm i personalizm; postmodernizm i dekonstrukcjonizm. 14. Wokół egzystencjalizm: Szestow, Heidegger, Jaspers, Sartre’a; lub filozofia życia: Nietzsche, Dilthey, Bergson. 15. Cum grano salis: wokół „partii matematycznej” w filozofii: o pitagorejczykach, Platonie, Kartezjuszu, Spinozie, Kancie, Whiteheadzie, Russellu... pod kątem ich przekonania o możliwość wykorzystania rozwiązań matematycznych do rozstrzygnięć ontologicznych i metafizycznych, jak również etycznych i politycznych. 								
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 913 1422 1070"> <tr> <td data-bbox="225 913 1161 1003">Posiada podstawową wiedzę o specyfice i roli filozofii oraz podstawowych zmianach w obrębie poruszanej przez nią problematyki od starożytności do współczesności</td> <td data-bbox="1166 913 1422 1003">K_W09</td> </tr> <tr> <td data-bbox="225 1003 1161 1070">Charakteryzuje różne - omawiane na ćwiczeniach i wykładzie - koncepcje filozoficzne; wymienia najbardziej charakterystyczne dla nich założenia</td> <td data-bbox="1166 1003 1422 1070">K_W09</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 1160 1422 1256"> <tr> <td data-bbox="225 1160 1161 1256">Potrafi analizować tekst, a także interpretować problemy w nim zawarte w szerszym tle odniesień do kultury współczesnej i własnej wrażliwości Stosuje poprawnie podstawowe pojęcia z zakresu filozofii</td> <td data-bbox="1166 1160 1422 1256"></td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 1339 1422 1429"> <tr> <td data-bbox="225 1339 1161 1429">Przejawia otwartość i wrażliwość na różnego rodzaju argumentacje i dostrzega ograniczenia różnego typu argumentacji, zwłaszcza tych które uznaje za optymalne</td> <td data-bbox="1166 1339 1422 1429"></td> </tr> </table>	Posiada podstawową wiedzę o specyfice i roli filozofii oraz podstawowych zmianach w obrębie poruszanej przez nią problematyki od starożytności do współczesności	K_W09	Charakteryzuje różne - omawiane na ćwiczeniach i wykładzie - koncepcje filozoficzne; wymienia najbardziej charakterystyczne dla nich założenia	K_W09	Potrafi analizować tekst, a także interpretować problemy w nim zawarte w szerszym tle odniesień do kultury współczesnej i własnej wrażliwości Stosuje poprawnie podstawowe pojęcia z zakresu filozofii		Przejawia otwartość i wrażliwość na różnego rodzaju argumentacje i dostrzega ograniczenia różnego typu argumentacji, zwłaszcza tych które uznaje za optymalne	
Posiada podstawową wiedzę o specyfice i roli filozofii oraz podstawowych zmianach w obrębie poruszanej przez nią problematyki od starożytności do współczesności	K_W09								
Charakteryzuje różne - omawiane na ćwiczeniach i wykładzie - koncepcje filozoficzne; wymienia najbardziej charakterystyczne dla nich założenia	K_W09								
Potrafi analizować tekst, a także interpretować problemy w nim zawarte w szerszym tle odniesień do kultury współczesnej i własnej wrażliwości Stosuje poprawnie podstawowe pojęcia z zakresu filozofii									
Przejawia otwartość i wrażliwość na różnego rodzaju argumentacje i dostrzega ograniczenia różnego typu argumentacji, zwłaszcza tych które uznaje za optymalne									

16.	<p>Literatura obowiązkowa i zalecana</p> <p>Literatura obowiązkowa do ćwiczeń:</p> <ol style="list-style-type: none"> 1. Obraz filozofii i filozofa: „Obrona Sokratesa” Platona 2. Wokół filozofii starożytnego świata: Wybrane fragmenty z „Państwa” Platona (rozdział 7) i „Metafizyki” Arystotelesa z ksiąg II i IV 3. Wokół filozofii średniowiecznego świata: Wybrane fragmenty z Augustyna „Soliloquium” (rozdział 2,6) i Tomasza „O królowaniu” (rozdział 13-15) 4. Dylematy nowożytności: Wybrany - przez studentów - jeden blok: 1.) „Rozprawy o metodzie” Descartes’a i „Etyki” Spinozy lub 2.) Pascala „Myśli” lub 3.) Wolter „Kandyd, czyli optymizm” lub 4.) Locke „List o tolerancji” 5. Systemy nowożytnego świata: Wybrany - przez studentów - jeden blok: 1.) fragmenty z „Monadologii” Leibniza lub 2.) fragmenty z „Prolegomeny” Kanta lub 3.) fragmenty z „Nauki logiki” Hegla 6. Różne oblicza współczesności: Wybrany - przez studentów - jeden blok 1.) Kuhn „Struktura rewolucji naukowych” (roz. V „Priorytet paradygmatów” lub 2.) Fromm „Ucieczka od wolności” (roz. IV „Dwa aspekty wolności człowieka współczesnego”) lub 3.) J. P. Sartre „Byt i Nicość” (Wprowadzenie: W poszukiwaniu bytu) lub 4.) Th. W. Adorno „Teoria estetyczna” („Wczesne wprowadzenie”) 7. O filozofii, nauce, matematyce i życiu na wesoło: H. Steinhaus „Słownik racjonalny” Literatura uzupełniająca (do wykładu) wybrany podręcznik: <ul style="list-style-type: none"> • W. Tatarkiewicz „Historia filozofii” t. 1-3 • S. Swieżawski „Dzieje europejskiej filozofii klasycznej” • B. Russell „Dzieje filozofii zachodu” 												
17.	Metody weryfikacji zakładanych efektów uczenia się egzamin ustny												
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu egzamin ustny weryfikujący opanowanie wiedzy i umiejętności.												
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="220 1133 1426 1171"> <tr> <td>wykład</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="220 1245 1426 1346"> <tr> <td>studiowanie tematyki wykładów i literatury</td> <td>30 godz.</td> </tr> <tr> <td>przygotowanie do egzaminu</td> <td>10 godz.</td> </tr> <tr> <td>przygotowanie raportu/prezentacji</td> <td>10 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="220 1429 1426 1496"> <tr> <td>Łączna liczba godzin</td> <td>80 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>3</td> </tr> </table>	wykład	30 godz.	studiowanie tematyki wykładów i literatury	30 godz.	przygotowanie do egzaminu	10 godz.	przygotowanie raportu/prezentacji	10 godz.	Łączna liczba godzin	80 godz.	Liczba punktów ECTS	3
wykład	30 godz.												
studiowanie tematyki wykładów i literatury	30 godz.												
przygotowanie do egzaminu	10 godz.												
przygotowanie raportu/prezentacji	10 godz.												
Łączna liczba godzin	80 godz.												
Liczba punktów ECTS	3												

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Idee post-humanizmu: projekt sztucznej inteligencji i problem możliwych umysłów Ideas of Post-Humanism: the Artificial Intelligence Project and the Problem of Possible Minds
2.	Dyscyplina filozofia
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Politologii
5.	Kod przedmiotu/modułu 28-INF-S-DHSPostHum
6.	Rodzaj przedmiotu/modułu Humanistyczno-społeczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest zaznajomienie słuchaczy z podstawowymi ideami określającymi nurt post-humanizmu, który pojawił się wraz z programem naturalizacji humanistyki. W tym podstaw rozumienia kultury, relacji społecznych umysłu i świadomości w kontekście różnych nurtów nauk kognitywnych.
14.	Treści programowe <ol style="list-style-type: none"> 1. Metafora komputera w badaniach nad umysłem i świadomością oraz jej konsekwencje dla rozumienia kultury i człowieka. 2. Post-humanizm czy humanizm rozszerzony – komu lub czemu możemy przypisać cechę umysłu. 3. Kultura jako rzeczywistość wirtualna: problem znaczenia, sensu i komunikacji w teorii informacji. 4. Humanistyka jako artefakt kultury w świecie technologii informacyjnych. 5. Mózg Boltzmanna i inne eksperymenty myślowe czyli rzeczywistość z punktu widzenia możliwego umysłu. 6. Umysł świadomy czyli jaki? Stopniowalność świadomości i idea panpsychizmu. 7. Czym jest świadomość – symulatorem czy symulacją: pomieszanie porządków logicznych i przy czynowo – skutkowych. 8. Zakres pojęcia świadomości: perspektywa pierwszo-osobowa i perspektywa trzecio-osobowa. Świat prywatnych fenomenów a rzeczywistość obiektywna i społeczna. 9. Świadomość refleksyjna i jej rola w procesach przetwarzania informacji o świecie. Problem podmiotu w naukach kognitywnych. 10. Idea odwróconej inżynierii: jeśli potrafimy skonstruować świadomy umysł to wiemy, czym on jest. 11. Świadomość i tożsamość: czy jesteśmy tym, kim myślimy, że jesteśmy? 12. Wielorakość tożsamości graczy i aktorów: świat społeczny jako symulacja ról. 13. Idea persony w klasycznej humanistyce i jako rola do odegrania. 14. Umysł i świadomość poza ludzkim mózgiem: problem A.I. i G.A.I. Systemy przetwarzania informacji i problem kontroli nad zachowaniami społecznymi. Dystopia końca ludzkości.

15.	Zakładane efekty uczenia się Wiedza zna podstawowe idee określające nurt post-humanizmu zna podstawy rozumienia kultury, relacji społecznych umysłu i świadomości w kontekście różnych nurtów nauk kognitywnych	K_W09 K_W09												
16.	Literatura obowiązkowa i zalecana Literatura obowiązkowa: <ol style="list-style-type: none"> 1. Bostrom N. (2016). Superinteligencja. Scenariusze, strategie, zagrożenia, Warszawa: Helion. 2. Chalmers D. (2010). Świadomy umysł, Warszawa: PWN. 3. Clark A., Chalmers D.J. (2008). Umysł rozszerzony, w: Analityczna metafizyka umysłu. Najnowsze kontrowersje, M. Miłkowski, R. Poczobut, Warszawa: IFiS PAN. 4. Dennett C.D. (2005). Słodkie sny. Filozoficzne przeszkody na drodze do nauki o świadomości, Poznań: Prószyński i S-ka. 5. Gazzaniga M.S. (2011). Istota człowieczeństwa. Co sprawia że jesteśmy wyjątkowymi, Sopot: Smak słowa, cz. II, W świecie społecznym. 6. Gleick J., Informacja. Bit, wszechświat, rewolucja, Kraków 2012. 7. Marcus G. Prowizorka w mózgu, Sopot: Smak słowa, rodz. 2, 4, 5, Pamięć, Dokonywanie wyborów, Język 8. Searle J.R. (2010). Umysł. Krótkie wprowadzenie, Poznań: Rebis, rozdz. 6, 9, 11, Intencjonalność, Nieświadomość i wyjaśnianie zachowania, Ja. Literatura zalecana <ul style="list-style-type: none"> • Damasio A.R. (2000). Tajemnica i świadomość. Jak ciało i emocje współtworzą świadomość, Poznań: Rebis. • Dennett D.C., Od bakterii do Bacha. O ewolucji umysłów, tłum. K. Bielecka, M. Miłkowski, Copernicus Center Press, Warszawa 2017. • Dennett D.C., Dźwignie wyobraźni i inne narzędzia do myślenia, tłum. Ł. Kurek, Copernicus Center Press, Warszawa 2015. • Gamble Clive, J. Gowlett, R. Dunbar, Potęga mózgu. Jak ewolucja życia społecznego kształtowała ludzki umysł, tłum. R. Kosarzycki, Copernicus Center Press, Kraków 2017. • Godfrey-Smith Peter, Inne umysły. Ośmiornice i prapoczątki świadomości, Copernicus Center Press, Kraków 2019 • Majchrowicz B., J. Doboszewski, T. Placek, Droga donikąd? Co pozostało z eksperymentów Libeta nad wolną wolą, w: „Filozofia nauki”, Rok XXIV, 2016, nr 2(94). 													
17.	Metody weryfikacji zakładanych efektów uczenia się udział w dyskusji, praca semestralna, egzamin													
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu obecność na ćwiczeniach, praca semestralna, egzamin													
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table border="1" data-bbox="225 1563 1166 1630"> <tr> <td>wykład</td> <td>30 godz.</td> </tr> <tr> <td>ćwiczenia</td> <td>30 godz.</td> </tr> </table> Praca własna studenta <table border="1" data-bbox="225 1709 1166 1776"> <tr> <td>studiowanie tematyki wykładów i literatury</td> <td>40 godz.</td> </tr> <tr> <td>przygotowanie do egzaminu</td> <td>20 godz.</td> </tr> </table> Sumarycznie <table border="1" data-bbox="225 1854 1166 1917"> <tr> <td>Łączna liczba godzin</td> <td>120 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>5</td> </tr> </table>	wykład	30 godz.	ćwiczenia	30 godz.	studiowanie tematyki wykładów i literatury	40 godz.	przygotowanie do egzaminu	20 godz.	Łączna liczba godzin	120 godz.	Liczba punktów ECTS	5	
wykład	30 godz.													
ćwiczenia	30 godz.													
studiowanie tematyki wykładów i literatury	40 godz.													
przygotowanie do egzaminu	20 godz.													
Łączna liczba godzin	120 godz.													
Liczba punktów ECTS	5													

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Manipulacje i nadużycia na rynku finansowym Manipulation and abuses on the financial market
2.	Dyscyplina ekonomia i finanse
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Nauk Ekonomicznych
5.	Kod przedmiotu/modułu 28-INF-S-DHSManINad
6.	Rodzaj przedmiotu/modułu Ekonomiczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele kształcenia dla przedmiotu <ul style="list-style-type: none"> 1. Zapoznanie studenta z różnymi formami i postaciami szeroko rozumianych nadużyć i manipulacji na rynku finansowym w Polsce. 2. Wyposażenie studenta w wiedzę na temat przestępstw na rynku finansowym. 3. Zapoznanie studenta z różnego rodzaju nadużyciami instytucji finansowych w relacjach umownych z klientami (z niebezpiecznymi, strukturyzowanymi i skomplikowanymi produktami finansowymi o dużym ryzyku oraz nieuczciwymi praktykami rynkowymi wprowadzającymi w błąd, praktykami naruszającymi zbiorowe interesy konsumentów, missellingiem). 4. Wyposażenie studenta w wiedzę na temat zagrożeń wynikających z zastosowania nowych technologii oraz umiejętność ich diagnozowania i unikania ich negatywnych skutków.

14.	<p>Treści programowe</p> <ol style="list-style-type: none"> 1. Formy i przejawy szeroko rozumianych nadużyć i manipulacji na rynku finansowym w Polsce 2. Stosowanie niedozwolonych klauzul umownych w konsumenckich umowach o usługi finansowe jako zakazana praktyka konsumencka (decyzje Prezesa UOKIK w tych sprawach i nakładane kary) 3. Praktyki naruszające zbiorowe interesy konsumentów na rynku usług finansowych (decyzje Prezesa UOKIK w tych sprawach i nakładane kary) 4. Nieuczciwe praktyki rynkowe na rynku usług finansowych. Manipulowanie informacją o odpłatności za usługi finansowe 5. Czyny nieuczciwej konkurencji na rynku finansowym 6. Nadużycia w relacjach umownych instytucji finansowych z konsumentami (problem niebezpiecznych, hybrydowych, strukturyzowanych produktów finansowych zagrożonych dużym ryzykiem, polisolokaty, „kredyty frankowe”, rabatowanie kosztów kredytu dla konsumentów) 7. Misselling na rynku usług finansowych 8. Nadużycia związane w przeprowadzaniem transakcji płatniczych (omyłkowy przelew na złe konto, procedura zwrotu pieniędzy, ataki phishingowe na klientów banku, wykonanie transakcji nieautoryzowanej, nienależyte wykonanie transakcji) 9. Nowe technologie na rynku finansowym i problem cyberbezpieczeństwa. Kradzież tożsamości i inne fraudy. 10. Zjawisko prania brudnych pieniędzy oraz mechanizmy przeciwdziałania 11. Nadużycia na rynku kapitałowym (manipulacje instrumentami finansowymi i insider trading), nadużycia na rynku obligacji 12. Prowadzenie działalności na rynku finansowym bez zezwolenia lub wpisu do rejestru 13. Piramidy finansowe 14. Przestępstwa na rynku finansowym w ujęciu ogólnym wynikające z przepisów karnych ustaw sektorowych 15. Przeciwdziałanie nadużyciom na rynku finansowym i mechanizmy usuwania ich skutków (mechanizmy karne, cywilnoprawne, administracyjnoprawne). Mechanizmy rozpatrywania reklamacji i sporów (dochodzenie indywidualnych roszczeń konsumenckich wynikających z nadużyć) 16. Ochrona konsumentów przed nadużyciami na rynku finansowym i rola organów ochrony. 														
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%; padding: 5px;">zna różne formy i postacie nadużyć na rynku finansowym</td> <td style="width: 30%; padding: 5px;">K_W09, Inż_W02</td> </tr> <tr> <td style="padding: 5px;">zna zachowania nielegalne instytucji finansowych oraz podmiotów podejmujących działalność na tym rynku bez wymaganego zezwolenia</td> <td style="padding: 5px;">K_W09, Inż_W02</td> </tr> <tr> <td style="padding: 5px;">zna, definiuje różne formy naruszeń instytucji finansowych wobec klientów w postaci nieuczciwych praktyk rynkowych, praktyk naruszających zbiorowe interesy konsumentów, missellingu, stosowania klauzul abuzywnych</td> <td style="padding: 5px;">K_W09, Inż_W02</td> </tr> </table> <p>Umiejętności</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%; padding: 5px;">potrafi zdiagnozować zagrożenia związane z manipulowaniem informacją na rynku finansowym i określać charakter tych naruszeń i ich skutki prawne</td> <td style="width: 30%; padding: 5px;">Inż_U03</td> </tr> <tr> <td style="padding: 5px;">posługuje się aparatem pojęciowym w zakresie nadużyć i manipulacji na rynku finansowym</td> <td style="padding: 5px;">Inż_U03</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%; padding: 5px;">potrafi identyfikować niebezpieczne, skomplikowane produkty finansowe</td> <td style="width: 30%;"></td> </tr> <tr> <td style="padding: 5px;">zna różne formy przestępstw i innych nadużyć na rynku finansowym</td> <td></td> </tr> </table>	zna różne formy i postacie nadużyć na rynku finansowym	K_W09, Inż_W02	zna zachowania nielegalne instytucji finansowych oraz podmiotów podejmujących działalność na tym rynku bez wymaganego zezwolenia	K_W09, Inż_W02	zna, definiuje różne formy naruszeń instytucji finansowych wobec klientów w postaci nieuczciwych praktyk rynkowych, praktyk naruszających zbiorowe interesy konsumentów, missellingu, stosowania klauzul abuzywnych	K_W09, Inż_W02	potrafi zdiagnozować zagrożenia związane z manipulowaniem informacją na rynku finansowym i określać charakter tych naruszeń i ich skutki prawne	Inż_U03	posługuje się aparatem pojęciowym w zakresie nadużyć i manipulacji na rynku finansowym	Inż_U03	potrafi identyfikować niebezpieczne, skomplikowane produkty finansowe		zna różne formy przestępstw i innych nadużyć na rynku finansowym	
zna różne formy i postacie nadużyć na rynku finansowym	K_W09, Inż_W02														
zna zachowania nielegalne instytucji finansowych oraz podmiotów podejmujących działalność na tym rynku bez wymaganego zezwolenia	K_W09, Inż_W02														
zna, definiuje różne formy naruszeń instytucji finansowych wobec klientów w postaci nieuczciwych praktyk rynkowych, praktyk naruszających zbiorowe interesy konsumentów, missellingu, stosowania klauzul abuzywnych	K_W09, Inż_W02														
potrafi zdiagnozować zagrożenia związane z manipulowaniem informacją na rynku finansowym i określać charakter tych naruszeń i ich skutki prawne	Inż_U03														
posługuje się aparatem pojęciowym w zakresie nadużyć i manipulacji na rynku finansowym	Inż_U03														
potrafi identyfikować niebezpieczne, skomplikowane produkty finansowe															
zna różne formy przestępstw i innych nadużyć na rynku finansowym															

16.	<p>Literatura obowiązkowa i zalecana</p> <p>Literatura Podstawowa</p> <ol style="list-style-type: none"> 1. Raport Doradczego Komitetu Naukowego przy Rzeczniku Finansowym: NIEPRAWIDŁOWOŚCI NA RYNKU FINANSOWYM A OCHRONA KONSUMENTA, Warszawa, wrzesień 2019 r., Doradczy Komitet Naukowy Rzecznika Finansowego, https://rf.gov.pl/pdf/DKN_Raport_nieprawidlowosci_wrzesien2019.pdf. 2019 2. Manipulacje i oszustwa na rynku finansowym. Perspektywa konsumenta. Wykrywanie, przeciwdziałanie, zapobieganie, Monkiewicz J., Jurkowska-Zeidler A., Rutkowska-Tomaszewska E., Wiktorow A. , Wydawnictwo Naukowe PWN. Warszawa 2020 3. Rynek finansowy. Zapobieganie przyczynom przestępczości (PDF), Blicharz G., Oręziak B., Wielec M., WYDAWNICTWO INSTYTUTU WYMIARU SPRAWIEDLIWOŚCI. Warszawa 2021 <p>Literatura Uzupełniająca</p> <ol style="list-style-type: none"> 1. Wybrane przestępstwa na rynku finansowym , Jaroszewicz P., Wojciechowska-Mytych K., Komisja Nadzoru Finansowego. Warszawa 2015 2. Manipulacja instrumentami finansowymi i insider trading. Analiza prawo-ekonomiczna, Martysz Cz., Wolters Kluwer Polska. Warszawa 2015 3. Obowiązki banków wobec klientów przewidziane w ustawie o obrocie instrumentami finansowymi, Majewski K., KNF. Warszawa 2019 4. OTWARTA BANKOWOŚĆ W ŚWIETLE WYMOGÓW DYREKTYWY PSD2 – wyzwania i perspektywy rozwoju dla polskiego sektora FinTech, (PDF), Leżoń K., KNF . Warszawa 2019 5. POŚREDNICY KREDYTOWI W ZAKRESIE KREDYTU KONSUMENCKIEGO ORAZ INSTYTUCJE POŻYCZKOWE – FORMALNE I PRAKTYCZNE ASPEKTY REJESTRACJI .Poradnik dla przedsiębiorcy i konsumenta (PDF), Brand. A., Przybysz M., Sakowska A., Świdarska J., KNF. Warszawa 2020 6. Whistleblowing w bankach, Cichy Ł., KNF. Warszawa 2017 7. Bezpieczeństwo konsumentów: na rynkach usług finansowych i społecznych, Kasprzak R.,Lubowiecki-Vikuk A. (red.), Oficyna Wydawnicza SGH. Warszawa 2020 8. Manipulacja i wykorzystanie przewagi informacyjnej na rynku finansowym, Więckowska M., Borowski K., Oficyna Wydawnicza SGH–Szkoła Główna Handlowa. Warszawa 2020 9. Misselling usług finansowych: problemy i uwarunkowania prawno-ekonomiczne, Nizioł K., Wydawnictwo Naukowe Uniwersytetu Szczecińskiego. Szczecin 2019 10. Piramidy finansowe: teoria, regulacje, praktyka, Masiukiewicz M., Wydawnictwo Naukowe PWN SA. Warszawa 2015 														
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>Egzamin, obecność na zajęciach, kolokwium zaliczeniowe</p>														
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <ul style="list-style-type: none"> • Wykład — egzamin • Ćwiczenia — praca końcowa semestralna, kolokwium opisowe 														
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 1541 1422 1608"> <tr> <td>wykład</td> <td>30 godz.</td> </tr> <tr> <td>ćwiczenia</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 1686 1422 1787"> <tr> <td>Przygotowanie do kolokwium</td> <td>20 godz.</td> </tr> <tr> <td>Przygotowanie do egzaminu</td> <td>20 godz.</td> </tr> <tr> <td>Czytanie zadanej literatury</td> <td>30 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 1865 1422 1935"> <tr> <td>Łączna liczba godzin</td> <td>130 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>5</td> </tr> </table>	wykład	30 godz.	ćwiczenia	30 godz.	Przygotowanie do kolokwium	20 godz.	Przygotowanie do egzaminu	20 godz.	Czytanie zadanej literatury	30 godz.	Łączna liczba godzin	130 godz.	Liczba punktów ECTS	5
wykład	30 godz.														
ćwiczenia	30 godz.														
Przygotowanie do kolokwium	20 godz.														
Przygotowanie do egzaminu	20 godz.														
Czytanie zadanej literatury	30 godz.														
Łączna liczba godzin	130 godz.														
Liczba punktów ECTS	5														

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Metody implementacji algorytmów Efficient Implementation of Algorithms
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DPMIA
6.	Rodzaj przedmiotu/modułu Inne
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin ćwiczenia — 15 godzin, pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zalecana znajomość przedmiotów <ul style="list-style-type: none"> • Wstęp do informatyki • Algorytmy i struktury danych Niezbędne kompetencje <ul style="list-style-type: none"> • Znajomość podstawowych algorytmów • Znajomość dowolnego wysokopoziomowego języka programowania (najlepiej C++)
13.	Cele kształcenia dla przedmiotu Celem przedmiotu są między innymi: <ul style="list-style-type: none"> • nauczenie studentów zarówno podstawowych, jak i bardziej zaawansowanych algorytmów, • przećwiczenie efektywnej implementacji algorytmów, • poznanie struktur danych przydatnych w pisaniu programów, • nauczenie prezentacji własnych rozwiązań, • zdobycie umiejętności analizowania problemów algorytmicznych, • zdobycie umiejętności wykorzystania znanych algorytmów przy rozwiązywaniu różnorodnych problemów, • przygotowanie do startu w zawodach algorytmicznych, tj. AMPPZ, CERC itd.

14.	<p>Treści programowe</p> <p>Rozwiązywanie problemów z zakresu algorytmów:</p> <ul style="list-style-type: none"> • zachłannych, • wymagających złożonej implementacji, • wyszukiwania binarnego, • dynamicznych, • struktury zbiorów rozłącznych, • struktury drzew przedziałowych, • kombinatorycznych, • teorii liczb, • gier, • grafowych, • skojarzeń, • maksymalnego przepływu, • tekstowych, • geometrycznych. 																								
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 719 1428 913"> <tr> <td data-bbox="225 719 1166 786">zna podstawowe metody projektowania, analizowania i implementowania algorytmów, a także podstawowe struktury danych i operacje na nich</td> <td data-bbox="1171 719 1428 786">K_W06</td> </tr> <tr> <td data-bbox="225 786 1166 819">ma wiedzę na temat języków programowania</td> <td data-bbox="1171 786 1428 819">K_W05</td> </tr> <tr> <td data-bbox="225 819 1166 853">zna i rozumie podstawowe techniki dowodzenia poprawność rozwiązań</td> <td data-bbox="1171 819 1428 853">K_W08</td> </tr> <tr> <td data-bbox="225 853 1166 913">ma podstawową wiedzę na temat matematyki, prawdopodobieństwa, matematyki dyskretnej, teorii liczb, teorii grafów i kombinatoryki</td> <td data-bbox="1171 853 1428 913">K_W03</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 996 1428 1415"> <tr> <td data-bbox="225 996 1166 1030">potrafi pisać, uruchamiać i testować programy</td> <td data-bbox="1171 996 1428 1030">K_U03</td> </tr> <tr> <td data-bbox="225 1030 1166 1093">potrafi projektować, implementować oraz analizować algorytmy pod kątem poprawności i złożoności obliczeniowej</td> <td data-bbox="1171 1030 1428 1093">K_U06</td> </tr> <tr> <td data-bbox="225 1093 1166 1189">potrafi ocenić przydatność typowych metod i narzędzi informatycznych oraz wybrać i zastosować odpowiednią metodę i narzędzia do typowych zadań informatycznych</td> <td data-bbox="1171 1093 1428 1189">K_U09</td> </tr> <tr> <td data-bbox="225 1189 1166 1254">potrafi prezentować opracowane zagadnienia, formułować opinie, a także podejmować dyskusję i analizować problemy informatyczne</td> <td data-bbox="1171 1189 1428 1254">K_U12</td> </tr> <tr> <td data-bbox="225 1254 1166 1350">potrafi posługiwać się podstawowymi pojęciami i twierdzeniami z zakresu matematyki, rachunku prawdopodobieństwa i statystyki oraz stosować je do rozwiązywania problemów</td> <td data-bbox="1171 1254 1428 1350">K_U04</td> </tr> <tr> <td data-bbox="225 1350 1166 1415">potrafi stosować poznane pojęcia i metody matematyki dyskretnej do modelowania i analizy problemów informatycznych</td> <td data-bbox="1171 1350 1428 1415">K_U04</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 1498 1428 1630"> <tr> <td data-bbox="225 1498 1166 1563">wykazuje samodzielność myślenia i działania przy rozwiązywaniu problemów i wykonywaniu zadań typowych dla zawodów związanych z informatyką</td> <td data-bbox="1171 1498 1428 1563">K_K06</td> </tr> <tr> <td data-bbox="225 1563 1166 1630">aktywnie prezentuje krytyczną postawę wobec stwierdzeń, uwag i wniosków, zwłaszcza nieopartych logicznym uzasadnieniem</td> <td data-bbox="1171 1563 1428 1630">K_K06</td> </tr> </table>	zna podstawowe metody projektowania, analizowania i implementowania algorytmów, a także podstawowe struktury danych i operacje na nich	K_W06	ma wiedzę na temat języków programowania	K_W05	zna i rozumie podstawowe techniki dowodzenia poprawność rozwiązań	K_W08	ma podstawową wiedzę na temat matematyki, prawdopodobieństwa, matematyki dyskretnej, teorii liczb, teorii grafów i kombinatoryki	K_W03	potrafi pisać, uruchamiać i testować programy	K_U03	potrafi projektować, implementować oraz analizować algorytmy pod kątem poprawności i złożoności obliczeniowej	K_U06	potrafi ocenić przydatność typowych metod i narzędzi informatycznych oraz wybrać i zastosować odpowiednią metodę i narzędzia do typowych zadań informatycznych	K_U09	potrafi prezentować opracowane zagadnienia, formułować opinie, a także podejmować dyskusję i analizować problemy informatyczne	K_U12	potrafi posługiwać się podstawowymi pojęciami i twierdzeniami z zakresu matematyki, rachunku prawdopodobieństwa i statystyki oraz stosować je do rozwiązywania problemów	K_U04	potrafi stosować poznane pojęcia i metody matematyki dyskretnej do modelowania i analizy problemów informatycznych	K_U04	wykazuje samodzielność myślenia i działania przy rozwiązywaniu problemów i wykonywaniu zadań typowych dla zawodów związanych z informatyką	K_K06	aktywnie prezentuje krytyczną postawę wobec stwierdzeń, uwag i wniosków, zwłaszcza nieopartych logicznym uzasadnieniem	K_K06
zna podstawowe metody projektowania, analizowania i implementowania algorytmów, a także podstawowe struktury danych i operacje na nich	K_W06																								
ma wiedzę na temat języków programowania	K_W05																								
zna i rozumie podstawowe techniki dowodzenia poprawność rozwiązań	K_W08																								
ma podstawową wiedzę na temat matematyki, prawdopodobieństwa, matematyki dyskretnej, teorii liczb, teorii grafów i kombinatoryki	K_W03																								
potrafi pisać, uruchamiać i testować programy	K_U03																								
potrafi projektować, implementować oraz analizować algorytmy pod kątem poprawności i złożoności obliczeniowej	K_U06																								
potrafi ocenić przydatność typowych metod i narzędzi informatycznych oraz wybrać i zastosować odpowiednią metodę i narzędzia do typowych zadań informatycznych	K_U09																								
potrafi prezentować opracowane zagadnienia, formułować opinie, a także podejmować dyskusję i analizować problemy informatyczne	K_U12																								
potrafi posługiwać się podstawowymi pojęciami i twierdzeniami z zakresu matematyki, rachunku prawdopodobieństwa i statystyki oraz stosować je do rozwiązywania problemów	K_U04																								
potrafi stosować poznane pojęcia i metody matematyki dyskretnej do modelowania i analizy problemów informatycznych	K_U04																								
wykazuje samodzielność myślenia i działania przy rozwiązywaniu problemów i wykonywaniu zadań typowych dla zawodów związanych z informatyką	K_K06																								
aktywnie prezentuje krytyczną postawę wobec stwierdzeń, uwag i wniosków, zwłaszcza nieopartych logicznym uzasadnieniem	K_K06																								
16.	<p>Literatura obowiązkowa i zalecana</p> <ul style="list-style-type: none"> • Cormen Thomas H., Leiserson Charles E., Rivest Ronald L, Clifford Stein, Wprowadzenie do algorytmów • blog na stronie https://codeforces.com/ • https://cp-algorithms.com/ • https://solve.edu.pl/sparingi/resources 																								
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>Rozwiązanie zadań poruszonych na zajęciach.</p>																								
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Do zaliczenia ćwiczenio-pracowni należy rozwiązać odpowiednią liczbę zadań na zajęciach, oraz wykazać odpowiednią aktywność na omówieniu zadań.</p> <p>Dodatkowa aktywność oraz prezentacje rozwiązań mogą wpłynąć na obniżenie powyższego limitu.</p>																								

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	ćwiczenia	15 godz.
	Praca własna studenta	
	dobrze zrozumienie oraz zaimplementowanie rozwiązań przedstawionych na zajęciach	60 godz.
	przejrzenie materiałów przekazanych przez prowadzących przed zajęciami	5 godz.
	Sumarycznie	
	Łączna liczba godzin	110 godz.
	Liczba punktów ECTS	4

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Ochrona własności intelektualnej Intellectual Property Rights
2.	Dyscyplina nauki prawne
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Prawa Cywilnego
5.	Kod przedmiotu/modułu 28-INF-S-KOWI
6.	Rodzaj przedmiotu/modułu Nieinformatyczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr nieokreślony
11.	Forma zajęć i liczba godzin wykład — 8 godzin, ćwiczenia — 7 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest kształcenie umiejętności praktycznej analizy przepisów dotyczących ochrony własności intelektualnej oraz świadomości konieczności ich przestrzegania.
14.	Treści programowe <ol style="list-style-type: none"> 1. Źródła praw na dobrach niematerialnych. 2. Zakres przedmiotowy i podmiotowy prawa autorskiego. 3. Autorskie prawa osobiste i majątkowe. 4. Okres ochrony utworu. 5. Domena publiczna. 6. Dozwolony użytek prywatny i publiczny. 7. Prawo cytatu i plagiat. 8. Odpowiedzialność za naruszenie praw autorskich. 9. Umowy o przekazaniu praw autorskich i umowy licencyjne. 10. Wolne licencje, Crative Commons i Ruch Wolnej Kultury. 11. Organizacje zbiorowego zarządzania prawami autorskimi i pokrewnymi. 12. Prawa własności przemysłowej. 13. Bazy danych. 14. Zwalczanie nieuczciwej konkurencji.

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 192 1161 259"> <tr> <td>Ma podstawową wiedzę na temat aktów prawnych obowiązujących w zakresie ochrony własności intelektualnej</td> <td>K_W10, Inż_W02</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 342 1161 472"> <tr> <td>potrafi analizować i interpretować tekst ustawy o prawie autorskim i prawach pokrewnych potrafi zaprezentować przygotowany materiał dotyczący wybranych zagadnień z zakresu ochrony własności intelektualnej</td> <td></td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 555 1161 622"> <tr> <td>ma świadomość konieczności przestrzegania prawa z zakresu ochrony własności intelektualnej</td> <td></td> </tr> </table>	Ma podstawową wiedzę na temat aktów prawnych obowiązujących w zakresie ochrony własności intelektualnej	K_W10, Inż_W02	potrafi analizować i interpretować tekst ustawy o prawie autorskim i prawach pokrewnych potrafi zaprezentować przygotowany materiał dotyczący wybranych zagadnień z zakresu ochrony własności intelektualnej		ma świadomość konieczności przestrzegania prawa z zakresu ochrony własności intelektualnej								
Ma podstawową wiedzę na temat aktów prawnych obowiązujących w zakresie ochrony własności intelektualnej	K_W10, Inż_W02													
potrafi analizować i interpretować tekst ustawy o prawie autorskim i prawach pokrewnych potrafi zaprezentować przygotowany materiał dotyczący wybranych zagadnień z zakresu ochrony własności intelektualnej														
ma świadomość konieczności przestrzegania prawa z zakresu ochrony własności intelektualnej														
16.	<p>Literatura obowiązkowa i zalecana</p> <ol style="list-style-type: none"> 1. Ustawa z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz.U. z 2017r. poz. 880) 2. Krótki kurs własności intelektualnej. Materiały dla uczelni http://prawokultury.pl/kurs/ 3. Marek Łazewski, Mariusz Gołębiowski: Vademecum Innowacyjnego Przedsiębiorcy, tom III – Własność intelektualna, Warszawa 2006, ISDN 83-907625-6-0. 4. Ustawa z dnia 30 czerwca 2000 r. Prawo własności przemysłowej (Dz.U. z 2017r. poz. 776, z późn. zm.) 5. Traktat Światowej Organizacji Własności Intelektualnej o Prawie Autorskim, sporządzony w Genewie dnia 20 grudnia 1996 r. (Dz.U. z 2005r. Nr 3, poz.12) 6. Richard M. Stallman: „Własność intelektualna” to zwodniczy miraż, 2004 													
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>przygotowanie pracy pisemnej dotyczącej zagadnienia z zakresu ochrony własności intelektualnej</p>													
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Warunkiem zaliczenia przedmiotu jest:</p> <ul style="list-style-type: none"> • obecność na zajęciach - dopuszczalna jest jedna nieobecność, którą należy zaliczyć w formie uzgodnionej z prowadzącym, • aktywny udział w grupowej analizie tekstu, • przedstawienie opracowanego zagadnienia z zakresu ochrony własności intelektualnej. 													
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 1406 1161 1473"> <tr> <td>wykład</td> <td>8 godz.</td> </tr> <tr> <td>ćwiczenia</td> <td>7 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 1556 1161 1624"> <tr> <td>studiowanie tematyki wykładów i literatury</td> <td>10 godz.</td> </tr> <tr> <td>przygotowanie raportu/prezentacji</td> <td>10 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 1706 1161 1774"> <tr> <td>Łączna liczba godzin</td> <td>35 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>1</td> </tr> </table>	wykład	8 godz.	ćwiczenia	7 godz.	studiowanie tematyki wykładów i literatury	10 godz.	przygotowanie raportu/prezentacji	10 godz.	Łączna liczba godzin	35 godz.	Liczba punktów ECTS	1	
wykład	8 godz.													
ćwiczenia	7 godz.													
studiowanie tematyki wykładów i literatury	10 godz.													
przygotowanie raportu/prezentacji	10 godz.													
Łączna liczba godzin	35 godz.													
Liczba punktów ECTS	1													

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Od Mordoru po Matrix. Fantastyczna filozofia polityki From Mordor to Matrix. The Fantastic Political Philosophy
2.	Dyscyplina nauki o polityce i administracji
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Politologii
5.	Kod przedmiotu/modułu 28-INF-S-DHSFantaFil
6.	Rodzaj przedmiotu/modułu Humanistyczno-społeczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu <ul style="list-style-type: none"> • Zainteresowanie fantastyką literacką i filmową jako filozoficzną i polityczną (ideologiczną) formą wypowiedzi, • Zainteresowanie myślą polityczną i filozofią polityki.
13.	Cele kształcenia dla przedmiotu Zapoznanie studentów z problematyką będącą przedmiotem zainteresowania filozofii oraz antropologii polityki (natura polityki, wspólnoty politycznej i człowieka jako „zwierzęcia politycznego”; uniwersalne mechanizmy i aporie władzy, koncepcje strukturalnej organizacji społeczeństwa; pochodzenie, zmienność, ewolucja– progres, regres, cykliczność - i postulowana wartość moralna ustrojów oraz instytucji politycznych; światopoglądy, ideologie, modele polityczne i ich zderzenia z praxis; legitymizacja; dyskurs polityczny; konflikt społeczny; rewolucja i wojna; mechanizmy fanatyzmu, ekstremizmu i radykalizmu, utopia jako fikcja spekulatywna; dystopia jako krytyka polityczna, sprzężenie zwrotne między nieświadomą i uświadomioną fikcją a praktyką polityczną) poprzez analizę utworów, gatunków, konwencji i toposów historycznej oraz współczesnej fantastyki.

14.	<p>Treści programowe</p> <ol style="list-style-type: none"> 1. Polityczne znaczenie mitów „starego” i „nowego” typu (refleksja historiozoficzna, premodernizm kontra nowoczesność; dialektyka Oświecenia, natura, kultura i technologia a człowieczeństwo); 2. Wpływ fantastyki (fikcji spekulatywnej) na politykę, religię oraz naukę i vice versa; 3. Utopia (futurotopia, retrotopia) i dystopia (ich treść, funkcje społeczne i polityczne, potencjał intelektualny oraz moralny); 4. Wizje bliskiej i odległej przyszłości ludzkości w kontekście polityczno-społecznym; 5. Rzeczywistość wirtualna i sztuczne inteligencje w kontekście politycznym. 6. Rewolucja biotechnologiczna, transhumanizm, Osobliwość („kryzys narzędziowni” i konieczność rewolucji paradygmatycznej). 7. Polityczne i antropologiczne konteksty, toposy, funkcje i znaczenia charakterystyczne dla form i gatunków (mit, utopia, fantasy, science fiction, cyberpunk, postapokaliptyka) oraz dzieł literatury i filmu (m.in. „Metropolis”, „The Shape of Things to Come”, „Odyseja kosmiczna 2001”, „Władca Pierścieni”, „Gwiezdne wojny”, „Żołnierze kosmosu”, „1984”, „Matrix”) i autorów (m.in. Morus, Wells, Orwell, Campbell, Lem, Dukaj, Bradbury, Robinson, Liu, Stephenson). 								
15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 689 1426 824"> <tr> <td data-bbox="225 689 1166 757">Zna kategorie filozoficzne występujące w analizowanych tekstach kultury oraz ich rolę w kształtowaniu refleksji politologicznej</td> <td data-bbox="1174 689 1426 757">K_W09</td> </tr> <tr> <td data-bbox="225 757 1166 824">Zna najważniejsze nurty myśli politycznej przejawiające się w analizowanych klasycznych i współczesnych dziełach fikcji spekulatywnej</td> <td data-bbox="1174 757 1426 824">K_W09</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 902 1426 1160"> <tr> <td data-bbox="225 902 1166 1160">Potrafi porządkować, klasyfikować, porównywać oraz analizować i interpretować zjawiska polityczne analizowane w wybranych klasycznych dziełach fikcji spekulatywnej, uwzględniając wzajemne oddziaływania zjawisk politycznych i społecznych Potrafi weryfikować proponowane w analizowanych dziełach fikcji spekulatywnej rozwiązania problemów politycznych. Potrafi poddawać krytyce proponowane rozwiązania i argumentować sądy na temat możliwych rozwiązań problemu</td> <td data-bbox="1174 902 1426 1160"></td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 1238 1426 1368"> <tr> <td data-bbox="225 1238 1166 1368">Jest gotów do krytycznej oceny swojej wiedzy w zakresie fikcji spekulatywnej oraz filozofii polityki, ciągłego doksztalcania się i rozwoju zawodowego oraz korzystania z opinii ekspertów w przypadku trudności z samodzielnym rozwiązaniem problemów</td> <td data-bbox="1174 1238 1426 1368"></td> </tr> </table>	Zna kategorie filozoficzne występujące w analizowanych tekstach kultury oraz ich rolę w kształtowaniu refleksji politologicznej	K_W09	Zna najważniejsze nurty myśli politycznej przejawiające się w analizowanych klasycznych i współczesnych dziełach fikcji spekulatywnej	K_W09	Potrafi porządkować, klasyfikować, porównywać oraz analizować i interpretować zjawiska polityczne analizowane w wybranych klasycznych dziełach fikcji spekulatywnej, uwzględniając wzajemne oddziaływania zjawisk politycznych i społecznych Potrafi weryfikować proponowane w analizowanych dziełach fikcji spekulatywnej rozwiązania problemów politycznych. Potrafi poddawać krytyce proponowane rozwiązania i argumentować sądy na temat możliwych rozwiązań problemu		Jest gotów do krytycznej oceny swojej wiedzy w zakresie fikcji spekulatywnej oraz filozofii polityki, ciągłego doksztalcania się i rozwoju zawodowego oraz korzystania z opinii ekspertów w przypadku trudności z samodzielnym rozwiązaniem problemów	
Zna kategorie filozoficzne występujące w analizowanych tekstach kultury oraz ich rolę w kształtowaniu refleksji politologicznej	K_W09								
Zna najważniejsze nurty myśli politycznej przejawiające się w analizowanych klasycznych i współczesnych dziełach fikcji spekulatywnej	K_W09								
Potrafi porządkować, klasyfikować, porównywać oraz analizować i interpretować zjawiska polityczne analizowane w wybranych klasycznych dziełach fikcji spekulatywnej, uwzględniając wzajemne oddziaływania zjawisk politycznych i społecznych Potrafi weryfikować proponowane w analizowanych dziełach fikcji spekulatywnej rozwiązania problemów politycznych. Potrafi poddawać krytyce proponowane rozwiązania i argumentować sądy na temat możliwych rozwiązań problemu									
Jest gotów do krytycznej oceny swojej wiedzy w zakresie fikcji spekulatywnej oraz filozofii polityki, ciągłego doksztalcania się i rozwoju zawodowego oraz korzystania z opinii ekspertów w przypadku trudności z samodzielnym rozwiązaniem problemów									

16.	<p>Literatura obowiązkowa i zalecana</p> <p>Obowiązkowa</p> <ul style="list-style-type: none"> • Frederick Jameson, Archeologie przyszłości. Pragnienie zwane utopią i inne fantazje naukowe, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2012. • James Gunn, Droga do science fiction, Wydawnictwo Alfa, 4 (1985-1988). • Tomasz Majkowski, W cieniu białego drzewa. Powieść fantasy w XX wieku, Wydawnictwo Uniwersytetu Jagiellońskiego 2013. <p>Zalecana</p> <ul style="list-style-type: none"> • The Cambridge Companion to Science Fiction (red. Edward James, Farah Mendlesohn), Cambridge University Press 2003. • The Oxford Handbook of Science Fiction (red. Rob Latham), Oxford University Press 2014. • Adam Roberts, The History of Science Fiction, Palgrave Macmillan 2016. • Reading Science Fiction (red. James Gunn, Marleen S. Barr, Matthew Candelaria), Palgrave Macmillan 2009. Carl Freedman, Critical Theory and Science Fiction, Wesleyan University Press 2000. • John Rieder, Colonialism and the Emergence of Science Fiction, Wesleyan University Press 2008. • The Philosophy of Science Fiction Film (red. Steven M. Sanders), The University Press of Kentucky 2008. • Patrick Curry, Defending Middle-Earth. Tolkien: Myth and Modernity, HarperCollins 2012. • Abigail E. Ruane, Patrick James, The International Relations of Middle-Earth. Learning from The Lord of the Rings, The University of Michigan Press 2012. • Dariusz Brzostek, Fikcja antropologiczna czy antropologia spekulatywna? O funkcjach poznawczych narracji fantastycznonaukowych, „Prace kulturoznawcze” 23, nr 2-3, Wrocław 2019. 										
17.	Metody weryfikacji zakładanych efektów uczenia się egzamin pisemny										
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu egzamin pisemny										
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">wykład</td> <td style="width: 20%;">30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">przygotowanie do egzaminu lub rozwiązywanie dodatkowych zadań</td> <td style="width: 20%;">20 godz.</td> </tr> <tr> <td>lektura tekstów</td> <td>20 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%;">Łączna liczba godzin</td> <td style="width: 20%;">70 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>3</td> </tr> </table>	wykład	30 godz.	przygotowanie do egzaminu lub rozwiązywanie dodatkowych zadań	20 godz.	lektura tekstów	20 godz.	Łączna liczba godzin	70 godz.	Liczba punktów ECTS	3
wykład	30 godz.										
przygotowanie do egzaminu lub rozwiązywanie dodatkowych zadań	20 godz.										
lektura tekstów	20 godz.										
Łączna liczba godzin	70 godz.										
Liczba punktów ECTS	3										

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Polityka gospodarcza Economic Policy
2.	Dyscyplina nauki o polityce i administracji
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Nauk Ekonomicznych
5.	Kod przedmiotu/modułu 28-INF-S-DHSPolGosp
6.	Rodzaj przedmiotu/modułu Humanistyczno-społeczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest zapoznanie studentów z podstawami i głównymi pojęciami polityki gospodarczej w otwartej gospodarce rynkowej, w celu umożliwienia im lepszego zrozumienia głównych zagadnień i problemów związanych z przebiegiem procesów gospodarczych.
14.	Treści programowe <ul style="list-style-type: none"> • Dziedzictwo gospodarki centralnie planowanej • Transformacja gospodarcza • Rozwój a wzrost gospodarczy • Polityka fiskalna • Polityka pieniężna • Rynek pracy • Problem starzenia się społeczeństw • Cykle koniunkturalne • Kryzysy gospodarcze • Inflacja • Ekonomiczne uwarunkowania procesów integracyjnych • Unia walutowa

15.	Zakładane efekty uczenia się	
	Wiedza	
	zna podstawowe i główne pojęcia polityki gospodarczej w otwartej gospodarce rynkowej	K_W09
	zna główne zagadnienie i problemy związane z kierowaniem procesami gospodarczymi	K_W09
	Umiejętności	
	potrafi analizować zjawiska i procesy zachodzące w gospodarce krajowej i międzynarodowej z wykorzystaniem standardowych metod i narzędzi stosowanych w ekonomii	
	potrafi ocenić rolę przedsiębiorczości jako „czwartego czynnika” ekonomicznego	
	potrafi samodzielnie interpretować zjawiska gospodarcze w powiązaniu z poznanymi koncepcjami teoretycznymi	
	potrafi ocenić wpływ uwarunkowań i ograniczeń natury politycznej, prawnej oraz społecznej na kształt realizowanej polityki ekonomicznej	
16.	Literatura obowiązkowa i zalecana	
	<ul style="list-style-type: none"> • B. Winiarski, Polityka gospodarcza, Wydawnictwo Naukowe PWN, Warszawa 2009 • T. Włudyka, Marcin Smaga, Instytucje gospodarki rynkowej, Wolters Kluwer, Warszawa 2018 	
17.	Metody weryfikacji zakładanych efektów uczenia się	
	Test obejmujący cały zakres wykładu	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Test jednokrotnego wyboru	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	30 godz.
	Praca własna studenta	
	Samodzielna praca studenta	50 godz.
	Sumarycznie	
	Łączna liczba godzin	80 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Praktyka programowania sportowego Sport programming in practice
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-PrProgSport
6.	Rodzaj przedmiotu/modułu Inne
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin pracownia — 75 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty • Metody implementacji algorytmów
13.	Cele kształcenia dla przedmiotu Podstawowym celem zajęć jest ułatwienie uczestnikom przygotowań do startów w zespołowych zawodach programistycznych (Akademickie Mistrzostwa Polski w Programowaniu Zespołowym, Central European Programming Contest, ICPC World Finals). Prowadzący (z pomocą uczestników) wybiera w każdym tygodniu pięciogodzinny konkurs o odpowiednim poziomie trudności. Następnie uczestnicy rozwiązują zadania z wybranego konkursu w tak zwanym trybie wirtualnym, czyli widząc jak ich aktualne wyniki mają się do wyników prawdziwych zawodów).
14.	Treści programowe Podczas zajęć uczestnicy rozwiązują zadania z zespołowych konkursów typu ICPC, w szczególności z wybranych regionalnych eliminacji do tych zawodów.

15.	Zakładane efekty uczenia się	
	Wiedza	
	zna podstawowe metody projektowania i analizy algorytmów	K_W07
	Umiejętności	
	potrafi formułować precyzyjną specyfikację problemu	K_U01
	potrafi sprawnie zaimplementować rozwiązanie zadania w odpowiednio wybranym języku programowania	K_U16
	potrafi sprawnie projektować, implementować oraz analizować algorytmy	K_U06
	potrafi efektywnie pracować w zespole	K_U14
Kompetencje społeczne		
	pracując z zespole dzieli zadania biorąc pod uwagę słabe i mocne strony każdego z zawodników w drużynie	K_K05
16.	Literatura obowiązkowa i zalecana Brak.	
17.	Metody weryfikacji zakładanych efektów uczenia się napisanie rozwiązań zadań z wybranych konkursów	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia przedmiotu potrzebny jest aktywny udział w zajęciach. Przez aktywny udział rozumie się rozwiązanie przynajmniej jednego zadania z konkursu wybranego na dany tydzień (podczas zajęć lub w terminie ustalonym i podanym na początku semestru razem z innymi zasadami).	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	75 godz.
	Sumarycznie	
	Łączna liczba godzin	75 godz.
Liczba punktów ECTS	2	

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Usługi finansowe Financial Services
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Nauk Ekonomicznych
5.	Kod przedmiotu/modułu 28-INF-S-DHSUslFin
6.	Rodzaj przedmiotu/modułu Ekonomiczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów ((jeśli obowiązuje)) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 30 godzin, ćwiczenia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele przedmiotu 1. Przedstawienie kompleksowej wiedzy o istocie, specyfice i cechach charakterystycznych usług finansowych oraz instytucji je świadczących (instytucji finansowych). 2. Zapoznanie studenta z zagrożeniami i ryzykiem związanym z usługami finansowymi. 3. Zapoznanie z całokształtem regulacji prawnych dotyczących usług finansowych.
14.	Treści programowe 1. Pojęcie i specyfika usług finansowych (usługi bankowe, kredytowe, płatnicze, ubezpieczeniowe, inwestycyjne); Klient usług finansowych (klient detaliczny, klient profesjonalny, konsument). Źródła zagrożeń i ryzyka dla klientów usług finansowych. Regulacja usług finansowych w UE. 2. Podmioty świadczące legalnie usługi finansowe w UE. Zasady podejmowania i prowadzenia działalności w zakresie świadczenia usług finansowych w UE. 3. Nowe technologie na rynku usług finansowych. Fin-Tech – problemy pojęciowe i regulacyjne. 4. Umowy o usługi finansowe i ich regulacja w UE - ogólna charakterystyka i tendencje dotyczące ich regulacji w UE. Koszty usług finansowych (Odpłatność za usługi finansowe). Odsetki, opłaty, prowizje i inne koszty ponoszone przez klientów). Obowiązki instytucji finansowych wobec klientów usług finansowych. 5. Usługi depozytowe ich charakterystyka i regulacja. 6. Usługi typu kredytowego ich charakterystyka i regulacja (kredyt bankowy, kredyt hipoteczny, kredyt konsumencki, konsumencki kredyt hipoteczny, odwrócony kredyt hipoteczny, leasing). 7. Usługi ubezpieczeniowe ich charakterystyka i regulacja. Dystrybucja usług ubezpieczeniowych. 8. Usługi płatnicze ich charakterystyka i regulacja. 9. Usługi inwestycyjne ich charakterystyka i regulacja. 10. Usługi emerytalne ich charakterystyka i regulacja.

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="223 190 1428 291"> <tr> <td data-bbox="223 190 1165 224">zna usługi finansowe skierowane do gospodarstw domowych (konsumentów)</td> <td data-bbox="1173 190 1428 224">K_W09, Inż_W02</td> </tr> <tr> <td data-bbox="223 224 1165 291">zna specyfikę każdego rodzaju usług finansowych (usługi bankowe, kredytowe, płatnicze, inwestycyjne, ubezpieczeniowe) oraz podmioty je świadczące</td> <td data-bbox="1173 224 1428 291">K_W09, Inż_W02</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="223 369 1428 470"> <tr> <td data-bbox="223 369 1165 403">potrafi wybrać usługę finansową adekwatną do jego potrzeb</td> <td data-bbox="1173 369 1428 403">Inż_U03</td> </tr> <tr> <td data-bbox="223 403 1165 470">potrafi wybrać usługę finansową dostosowaną do prowadzonej działalności gospodarczej i sytuacji finansowej</td> <td data-bbox="1173 403 1428 470">Inż_U03</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="223 548 1428 683"> <tr> <td data-bbox="223 548 1165 616">jest świadomy oczekiwań wynikających z korzystania z usług finansowych dla własnych potrzeb</td> <td data-bbox="1173 548 1428 616"></td> </tr> <tr> <td data-bbox="223 616 1165 683">potrafi ocenić usługi finansowe, które może zastosować we własnej działalności gospodarczej adekwatnie do potrzeb, sytuacji finansowej i sytuacji na rynku</td> <td data-bbox="1173 616 1428 683"></td> </tr> </table>	zna usługi finansowe skierowane do gospodarstw domowych (konsumentów)	K_W09, Inż_W02	zna specyfikę każdego rodzaju usług finansowych (usługi bankowe, kredytowe, płatnicze, inwestycyjne, ubezpieczeniowe) oraz podmioty je świadczące	K_W09, Inż_W02	potrafi wybrać usługę finansową adekwatną do jego potrzeb	Inż_U03	potrafi wybrać usługę finansową dostosowaną do prowadzonej działalności gospodarczej i sytuacji finansowej	Inż_U03	jest świadomy oczekiwań wynikających z korzystania z usług finansowych dla własnych potrzeb		potrafi ocenić usługi finansowe, które może zastosować we własnej działalności gospodarczej adekwatnie do potrzeb, sytuacji finansowej i sytuacji na rynku	
zna usługi finansowe skierowane do gospodarstw domowych (konsumentów)	K_W09, Inż_W02												
zna specyfikę każdego rodzaju usług finansowych (usługi bankowe, kredytowe, płatnicze, inwestycyjne, ubezpieczeniowe) oraz podmioty je świadczące	K_W09, Inż_W02												
potrafi wybrać usługę finansową adekwatną do jego potrzeb	Inż_U03												
potrafi wybrać usługę finansową dostosowaną do prowadzonej działalności gospodarczej i sytuacji finansowej	Inż_U03												
jest świadomy oczekiwań wynikających z korzystania z usług finansowych dla własnych potrzeb													
potrafi ocenić usługi finansowe, które może zastosować we własnej działalności gospodarczej adekwatnie do potrzeb, sytuacji finansowej i sytuacji na rynku													
16.	<p>Literatura obowiązkowa i zalecana</p> <p>Literatura Podstawowa</p> <ol data-bbox="247 784 1420 1321" style="list-style-type: none"> 1. Odwrócona hipoteka jako nowa usługa na rynku finansowym, E. Rutkowska-Tomaszewska (red.), C.H. Beck . Warszawa 2017, s. 15-38; 67-96; 187-216 2. Ochrona i edukacja konsumentów na rynku kredytów hipotecznych i konsumpcyjnych, Knehans-Olejnik A., CeDeWu. Warszawa 2015; s. 3-13 3. Konsument na rynku usług, Rosa G., C.H. Beck. Warszawa 2015; 198-222 4. Ochrona klienta na rynku usług finansowych w świetle aktualnych problemów i regulacji prawnych, E. Rutkowska- Tomaszewska (red.), C.H. Beck. Warszawa 2017 s. 3-98; 115-160; 267-388 5. Manipulowanie informacją w zakresie odpłatności za usługi bankowe i prawne mechanizmy przeciwdziałania tym nadużyciom, [w:] Nowe koncepcje i regulacje nadzoru finansowego, Rutkowska-Tomaszewska E., Kraków- Warszawa 2014, s. 297 6. Świat bankowości, Zaleska M. (red.), Difin. Warszawa 2018; s. 122-304 7. Innowacje cyfrowe w bankowości a włączenie cyfrowe i finansowe społeczeństwa / Mateusz Folwarski, Kraków: Wydawnictwo Uniwersytetu Jagiellońskiego, 2021. 8. Ochrona nieprofesjonalnego uczestnika rynku finansowego, Grażyna Szustak, Witold Gradoń, Łukasz Szewczyk, Katowice: Wydawnictwo Uniwersytetu Ekonomicznego w Katowicach, 2021. 9. Sektor FinTech na europejskim rynku usług bankowych: wyzwania konkurencyjne i regulacyjne / Mateusz Folwarski, Warszawa: Wydawnictwo Poltext, 2019. <p>Literatura Uzupełniająca</p> <ol data-bbox="247 1411 1420 2060" style="list-style-type: none"> 1. Informacja na rynku usług finansowych, Rutkowska- Tomaszewska E. (Red.), PWE. Warszawa 2019 2. Rynki finansowe. Organizacja, instytucje, uczestnicy, Banaszczak- Soroka U. (red.), CH Beck. Warszawa 2019 3. Sektor FinTech na europejskim rynku usług bankowych: wyzwania konkurencyjne i regulacyjne, Folwarski M., Poltext. Warszawa 2019. 4. Bankowość elektroniczna : istota i innowacje, Gospodarowicz A., CH Beck. Warszawa 2018. 5. Zrozumiałość przejrzystość i efektywność indywidualnych produktów emerytalnych w Polsce, Rutecka-Góra J., Bielawska K., Hadryan M., Kowalczyk-Rólczyńska P., Pieńkowska-Kamieniecka S., Wydawnictwo SGH. Warszawa 2020. 6. Usługi bankowości inwestycyjnej : wybrane zagadnienia, Cichy J., Gradoń W., Wydawnictwo Uniwersytetu Ekonomicznego w Katowicach. Katowice 2016. 7. Nowe technologie a sektor finansowy: Fintech jako szansa i zagrożenie, Szpringer W., Poltext. Warszawa 2017. 8. Private asset&mp; wealth management. Nowe instrumenty i usługi finansowe, Gabryleczyk K., C.H. Beck. Warszawa 2009. 9. Proces regulacji i deregulacji na rynku ubezpieczeniowym, płatniczym i kapitałowym, Byrski J., Szaraniec M., Magoń K., DIFIN . Warszawa 2019. 10. Instytucje bankowe i niebankowe na rynku detalicznych usług finansowych w Polsce, Waliszewski K., Czchowska I., CeDeWu. Warszawa 2019. 11. Bankowe terminowe umowy walutowe, Korpalski M., CH Beck. Warszawa 2020. 												

17.	Metody weryfikacji zakładanych efektów kształcenia Egzamin, kolokwium zaliczeniowe														
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu <ul style="list-style-type: none"> • Wykład - egzamin oraz obecność na zajęciach • Ćwiczenia – praca końcowa semestralna oraz kolokwium opisowe 														
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">wykład</td> <td style="width: 30%;">30 godz.</td> </tr> <tr> <td>ćwiczenia</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Czytanie zadanej literatury</td> <td style="width: 30%;">50 godz.</td> </tr> <tr> <td>Przygotowanie do kolokwium</td> <td>15 godz.</td> </tr> <tr> <td>Przygotowanie do egzaminu</td> <td>20 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Łączna liczba godzin</td> <td style="width: 30%;">145 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>5</td> </tr> </table>	wykład	30 godz.	ćwiczenia	30 godz.	Czytanie zadanej literatury	50 godz.	Przygotowanie do kolokwium	15 godz.	Przygotowanie do egzaminu	20 godz.	Łączna liczba godzin	145 godz.	Liczba punktów ECTS	5
wykład	30 godz.														
ćwiczenia	30 godz.														
Czytanie zadanej literatury	50 godz.														
Przygotowanie do kolokwium	15 godz.														
Przygotowanie do egzaminu	20 godz.														
Łączna liczba godzin	145 godz.														
Liczba punktów ECTS	5														

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Usługi finansowe (w) Financial Services (w)
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Nauk Ekonomicznych
5.	Kod przedmiotu/modułu 28-INF-S-DHSUslFin-W
6.	Rodzaj przedmiotu/modułu Ekonomiczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów ((jeśli obowiązuje)) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele przedmiotu 1. Przedstawienie kompleksowej wiedzy o istocie, specyfice i cechach charakterystycznych usług finansowych oraz instytucji je świadczących (instytucji finansowych). 2. Zapoznanie studenta z zagrożeniami i ryzykiem związanym z usługami finansowymi. 3. Zapoznanie z całokształtem regulacji prawnych dotyczących usług finansowych.
14.	Treści programowe 1. Pojęcie i specyfika usług finansowych (usługi bankowe, kredytowe, płatnicze, ubezpieczeniowe, inwestycyjne); Klient usług finansowych (klient detaliczny, klient profesjonalny, konsument). Źródła zagrożeń i ryzyka dla klientów usług finansowych. Regulacja usług finansowych w UE. 2. Podmioty świadczące legalnie usługi finansowe w UE. Zasady podejmowania i prowadzenia działalności w zakresie świadczenia usług finansowych w UE. 3. Nowe technologie na rynku usług finansowych. Fin-Tech – problemy pojęciowe i regulacyjne. 4. Umowy o usługi finansowe i ich regulacja w UE - ogólna charakterystyka i tendencje dotyczące ich regulacji w UE. Koszty usług finansowych (Odpłatność za usługi finansowe). Odsetki, opłaty, prowizje i inne koszty ponoszone przez klientów). Obowiązki instytucji finansowych wobec klientów usług finansowych. 5. Usługi depozytowe ich charakterystyka i regulacja. 6. Usługi typu kredytowego ich charakterystyka i regulacja (kredyt bankowy, kredyt hipoteczny, kredyt konsumencki, konsumencki kredyt hipoteczny, odwrócony kredyt hipoteczny, leasing). 7. Usługi ubezpieczeniowe ich charakterystyka i regulacja. Dystrybucja usług ubezpieczeniowych. 8. Usługi płatnicze ich charakterystyka i regulacja. 9. Usługi inwestycyjne ich charakterystyka i regulacja. 10. Usługi emerytalne ich charakterystyka i regulacja.

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 192 1426 293"> <tr> <td data-bbox="225 192 1166 226">zna usługi finansowe skierowane do gospodarstw domowych (konsumentów)</td> <td data-bbox="1174 192 1426 226">K_W09, Inż_W02</td> </tr> <tr> <td data-bbox="225 226 1166 293">zna specyfikę każdego rodzaju usług finansowych (usługi bankowe, kredytowe, płatnicze, inwestycyjne, ubezpieczeniowe) oraz podmioty je świadczące</td> <td data-bbox="1174 226 1426 293">K_W09, Inż_W02</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 371 1426 472"> <tr> <td data-bbox="225 371 1166 405">potrafi wybrać usługę finansową adekwatną do jego potrzeb</td> <td data-bbox="1174 371 1426 405">Inż_U03</td> </tr> <tr> <td data-bbox="225 405 1166 472">potrafi wybrać usługę finansową dostosowaną do prowadzonej działalności gospodarczej i sytuacji finansowej</td> <td data-bbox="1174 405 1426 472">Inż_U03</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 551 1426 683"> <tr> <td data-bbox="225 551 1166 683">jest świadomy oczekiwań wynikających z korzystania z usług finansowych dla własnych potrzeb potrafi ocenić usługi finansowe, które może zastosować we własnej działalności gospodarczej adekwatnie do potrzeb, sytuacji finansowej i sytuacji na rynku</td> <td data-bbox="1174 551 1426 683"></td> </tr> </table>	zna usługi finansowe skierowane do gospodarstw domowych (konsumentów)	K_W09, Inż_W02	zna specyfikę każdego rodzaju usług finansowych (usługi bankowe, kredytowe, płatnicze, inwestycyjne, ubezpieczeniowe) oraz podmioty je świadczące	K_W09, Inż_W02	potrafi wybrać usługę finansową adekwatną do jego potrzeb	Inż_U03	potrafi wybrać usługę finansową dostosowaną do prowadzonej działalności gospodarczej i sytuacji finansowej	Inż_U03	jest świadomy oczekiwań wynikających z korzystania z usług finansowych dla własnych potrzeb potrafi ocenić usługi finansowe, które może zastosować we własnej działalności gospodarczej adekwatnie do potrzeb, sytuacji finansowej i sytuacji na rynku	
zna usługi finansowe skierowane do gospodarstw domowych (konsumentów)	K_W09, Inż_W02										
zna specyfikę każdego rodzaju usług finansowych (usługi bankowe, kredytowe, płatnicze, inwestycyjne, ubezpieczeniowe) oraz podmioty je świadczące	K_W09, Inż_W02										
potrafi wybrać usługę finansową adekwatną do jego potrzeb	Inż_U03										
potrafi wybrać usługę finansową dostosowaną do prowadzonej działalności gospodarczej i sytuacji finansowej	Inż_U03										
jest świadomy oczekiwań wynikających z korzystania z usług finansowych dla własnych potrzeb potrafi ocenić usługi finansowe, które może zastosować we własnej działalności gospodarczej adekwatnie do potrzeb, sytuacji finansowej i sytuacji na rynku											
16.	<p>Literatura obowiązkowa i zalecana</p> <p>Literatura Podstawowa</p> <ol data-bbox="252 786 1420 1323" style="list-style-type: none"> 1. Odwrócona hipoteka jako nowa usługa na rynku finansowym, E. Rutkowska-Tomaszewska (red.), C.H. Beck . Warszawa 2017, s. 15-38; 67-96; 187-216 2. Ochrona i edukacja konsumentów na rynku kredytów hipotecznych i konsumpcyjnych, Knehans-Olejnik A., CeDeWu. Warszawa 2015; s. 3-13 3. Konsument na rynku usług, Rosa G., C.H. Beck. Warszawa 2015; 198-222 4. Ochrona klienta na rynku usług finansowych w świetle aktualnych problemów i regulacji prawnych, E. Rutkowska- Tomaszewska (red.), C.H. Beck. Warszawa 2017 s. 3-98; 115-160; 267-388 5. Manipulowanie informacją w zakresie odpłatności za usługi bankowe i prawne mechanizmy przeciwdziałania tym nadużyciom, [w:] Nowe koncepcje i regulacje nadzoru finansowego, Rutkowska-Tomaszewska E., Kraków- Warszawa 2014, s. 297 6. Świat bankowości, Zaleska M. (red.), Difin. Warszawa 2018; s. 122-304 7. Innowacje cyfrowe w bankowości a włączenie cyfrowe i finansowe społeczeństwa / Mateusz Folwarski, Kraków: Wydawnictwo Uniwersytetu Jagiellońskiego, 2021. 8. Ochrona nieprofesjonalnego uczestnika rynku finansowego, Grażyna Szustak, Witold Gradoń, Łukasz Szewczyk, Katowice: Wydawnictwo Uniwersytetu Ekonomicznego w Katowicach, 2021. 9. Sektor FinTech na europejskim rynku usług bankowych: wyzwania konkurencyjne i regulacyjne / Mateusz Folwarski, Warszawa: Wydawnictwo Poltext, 2019. <p>Literatura Uzupełniająca</p> <ol data-bbox="252 1413 1420 2069" style="list-style-type: none"> 1. Informacja na rynku usług finansowych, Rutkowska- Tomaszewska E. (Red.), PWE. Warszawa 2019 2. Rynki finansowe. Organizacja, instytucje, uczestnicy, Banaszczak- Soroka U. (red.), CH Beck. Warszawa 2019 3. Sektor FinTech na europejskim rynku usług bankowych: wyzwania konkurencyjne i regulacyjne, Folwarski M., Poltext. Warszawa 2019. 4. Bankowość elektroniczna : istota i innowacje, Gospodarowicz A., CH Beck. Warszawa 2018. 5. Zrozumiałość przejrzystość i efektywność indywidualnych produktów emerytalnych w Polsce, Rutecka-Góra J., Bielawska K., Hadryan M., Kowalczyk-Rólczyńska P., Pieńkowska-Kamieniecka S., Wydawnictwo SGH. Warszawa 2020. 6. Usługi bankowości inwestycyjnej : wybrane zagadnienia, Cichy J., Gradoń W., Wydawnictwo Uniwersytetu Ekonomicznego w Katowicach. Katowice 2016. 7. Nowe technologie a sektor finansowy: Fintech jako szansa i zagrożenie, Szpringer W., Poltext. Warszawa 2017. 8. Private asset& wealth management. Nowe instrumenty i usługi finansowe, Gabryleczyk K., C.H. Beck. Warszawa 2009. 9. Proces regulacji i deregulacji na rynku ubezpieczeniowym, płatniczym i kapitałowym, Byrski J., Szaraniec M., Magoń K., DIFIN . Warszawa 2019. 10. Instytucje bankowe i niebankowe na rynku detalicznych usług finansowych w Polsce, Waliszewski K., Czchowska I., CeDeWu. Warszawa 2019. 11. Bankowe terminowe umowy walutowe, Korpalski M., CH Beck. Warszawa 2020. 										

17.	Metody weryfikacji zakładanych efektów kształcenia Egzamin, kolokwium zaliczeniowe										
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu <ul style="list-style-type: none"> • Wykład - egzamin oraz obecność na zajęciach • Ćwiczenia – praca końcowa semestralna oraz kolokwium opisowe 										
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1"> <tr> <td>wykład</td> <td>30 godz.</td> </tr> </table> <p>Praca własna studenta</p> <table border="1"> <tr> <td>Czytanie zadanej literatury</td> <td>40 godz.</td> </tr> <tr> <td>Przygotowanie do egzaminu</td> <td>20 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1"> <tr> <td>Łączna liczba godzin</td> <td>90 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>3</td> </tr> </table>	wykład	30 godz.	Czytanie zadanej literatury	40 godz.	Przygotowanie do egzaminu	20 godz.	Łączna liczba godzin	90 godz.	Liczba punktów ECTS	3
wykład	30 godz.										
Czytanie zadanej literatury	40 godz.										
Przygotowanie do egzaminu	20 godz.										
Łączna liczba godzin	90 godz.										
Liczba punktów ECTS	3										

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Warsztaty: Jak założyć firmę? Workshop: making a start-up
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-WarJZF
6.	Rodzaj przedmiotu/modułu Inne
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin seminarium – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Niezbędne kompetencje <ul style="list-style-type: none"> • Znajomość podstawowych zasad inżynierii oprogramowania. • Zainteresowanie budowaniem własnej firmy
13.	Cele przedmiotu Celem zajęć jest przede wszystkim uświadomienie studentom, że praca w korporacji nie jest jedynym wyborem kariery zawodowej i że warto planować założenie własnej firmy. W tym celu poznają rodzaje działalności gospodarczej, sposoby finansowania start-upów i młodych firm, oraz rynek usług prawnych i finansowych. Ważnym celem przedmiotu jest przedstawienie przykładów konkretnych bardzo młodych firm oraz przykładów nieudanych decyzji biznesowych.
14.	Treści programowe <ul style="list-style-type: none"> • Co oferować: tanie usługi, unikalne usługi czy produkt. • Cel strategiczny: sprzedać firmę po zbudowaniu marki, czy budować z myślą o trwałej niezależnej firmie. • Jaki produkt: istniejący niezależnie, czy po zbudowaniu włączony do większego systemu. • Jak znaleźć niszę? • Jak ocenić wielkość rynku? • Czy myśleć o rynku lokalnym (Polska), czy globalnym? • Jak poszukiwać partnerów i klientów. • Finansowanie: gdzie są pieniądze? (prywatne i publiczne, granty dla start-upów). • Przykłady udanych startupów (w Polsce i za granicą) • Na ogół się nie udaje, co wtedy? • Firma rodzinna (koleżeńska), czy zatrudniająca liderów spoza grona znajomych? • Zrobimy to sami, czy kupimy usługę? • Czy zaczynać zaraz po studiach (lub jeszcze w trakcie), czy zdobyć wcześniej doświadczenie w korporacji? • Ile informatyki w produkcji (usłudze)? • Czy korzystać z usług księgowych, prawników, pijańców? Gdzie ich szukać? • Czy nawiązywać kontakty z sieciami społecznymi start-upowców.

15.	Zakładane efekty uczenia się	
	Wiedza	
	Posiada podstawową wiedzę o regulacjach prawnych dotyczących małych firm	K_W10
	Posiada podstawową wiedzę o ochronie własności intelektualnej	K_W10
	Zna zasady budowy "biznes planu"	K_W10
	Zna potencjalne źródła finansowania nowej firmy	K_W10
	Umiejętności	
	Potrafi opracować i zaprezentować zagadnienie dotyczące startupów	K_U12, K_U13
	Kompetencje społeczne	
	Usieciowienie - poznanie osób, które niedawno założyły firmy (głównie w branży IT)	K_K05
16.	Literatura obowiązkowa i zalecana	
17.	Metody weryfikacji zakładanych efektów kształcenia prezentacja zagadnienia, dyskusja	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Studenci przygotowują opracowanie (w formie krótkiego referatu będącego wprowadzeniem do dyskusji) jednego z tematów lub "case study" i udział w dyskusji.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	przygotowanie raportu/prezentacji	40 godz.
	Sumarycznie	
	Łączna liczba godzin	70 godz.
	Liczba punktów ECTS	3

9 Metasylabusy

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt zespołowy Group project
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu —
6.	Rodzaj przedmiotu/modułu Projekt
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr nieokreślony
11.	Forma zajęć i liczba godzin —
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu —
13.	Cele kształcenia dla przedmiotu <p>Podczas realizacji <i>projektu zespołowego</i> studenci uczą się uczestnictwa w dużym projekcie programistycznym, do którego przychodzą, gdy projekt jest już dość rozwinięty bądź rozpoczynając projekt od początku z myślą o późniejszym rozwoju projektu przez inne osoby. Studenci wykonują pewne zadania, zmagając się jednocześnie z niedoskonałościami, źródłem których jest fakt rozwijania projektu przez wiele osób. Jest to scenariusz bardziej odpowiadający realiom pracy zawodowej, niż na większości kursów, gdzie studenci tworzą w pojedynkę projekt „od zera”, który nie zawsze jest później rozwijany.</p>
14.	Treści programowe <ul style="list-style-type: none"> • Korzystanie ze standardowych narzędzi programistycznych: np. systemu kontroli wersji, CI (testów automatycznych). • Typowy proces tworzenia oprogramowania — zadania, pomysły, projektowanie, kodowanie, weryfikowanie (code review, testy), dołączanie.

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <table border="1" data-bbox="225 197 1161 232"> <tr> <td>Zna wybrane technologie używane w tworzeniu oprogramowania.</td> <td>K_W08, K_W10</td> </tr> </table> <p>Umiejętności</p> <table border="1" data-bbox="225 315 1161 696"> <tr> <td>Potrafi rozwijać, uruchamiać i testować oprogramowanie w zadanej technologii. Projektuje rozwiązanie dla postawionego zadania i prezentuje to rozwiązanie. Potrafi ocenić oraz uargumentować zasadność stosowania różnych narzędzi i technik.</td> <td>K_U03, Inż_U05 K_U03, K_U08, Inż_U05</td> </tr> <tr> <td>Potrafi zidentyfikować istotny dla siebie fragment kodu w dużym systemie informatycznym i przeanalizować jego działanie.</td> <td>K_U09, Inż_U01</td> </tr> <tr> <td>Umie używać popularnych narzędzi do pracy zespołu programistycznego (takich jak Slack, Github).</td> <td>K_U07, Inż_U02, Inż_U04</td> </tr> <tr> <td>Umie zrobić rzeczowe „code-review” i nie przyjmuje postawy defensywnej, gdy ktoś zrobi „code-review” jemu.</td> <td>K_U14, Inż_U02, Inż_U03, Inż_U04</td> </tr> <tr> <td>Umie korzystać z dokumentacji oraz znajdować w Internecie rozwiązania przy napotkanych przeszkodach technicznych.</td> <td>K_U13, Inż_U02, Inż_U04</td> </tr> </table> <p>Kompetencje społeczne</p> <table border="1" data-bbox="225 779 1161 1099"> <tr> <td>Potrafi pracować zespołowo przyjmując w grupie różne role; rozumie konieczność systematycznej pracy nad wszelkimi projektami, które mają długofalowy charakter.</td> <td>K_K01, K_K04</td> </tr> <tr> <td>Potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania.</td> <td>K_K03</td> </tr> <tr> <td>Potrafi myśleć i działać w sposób przedsiębiorczy.</td> <td>K_K06</td> </tr> <tr> <td>Pracując z zespołem dzieli zadania biorąc pod uwagę słabe i mocne strony każdego z członków zespołu.</td> <td>K_K05</td> </tr> <tr> <td>Rozumie i docenia znaczenie uczciwości intelektualnej w działaniach własnych i innych osób, postępuje etycznie.</td> <td>K_K01, K_K02, K_K03</td> </tr> </table>	Zna wybrane technologie używane w tworzeniu oprogramowania.	K_W08, K_W10	Potrafi rozwijać, uruchamiać i testować oprogramowanie w zadanej technologii. Projektuje rozwiązanie dla postawionego zadania i prezentuje to rozwiązanie. Potrafi ocenić oraz uargumentować zasadność stosowania różnych narzędzi i technik.	K_U03, Inż_U05 K_U03, K_U08, Inż_U05	Potrafi zidentyfikować istotny dla siebie fragment kodu w dużym systemie informatycznym i przeanalizować jego działanie.	K_U09, Inż_U01	Umie używać popularnych narzędzi do pracy zespołu programistycznego (takich jak Slack, Github).	K_U07, Inż_U02, Inż_U04	Umie zrobić rzeczowe „code-review” i nie przyjmuje postawy defensywnej, gdy ktoś zrobi „code-review” jemu.	K_U14, Inż_U02, Inż_U03, Inż_U04	Umie korzystać z dokumentacji oraz znajdować w Internecie rozwiązania przy napotkanych przeszkodach technicznych.	K_U13, Inż_U02, Inż_U04	Potrafi pracować zespołowo przyjmując w grupie różne role; rozumie konieczność systematycznej pracy nad wszelkimi projektami, które mają długofalowy charakter.	K_K01, K_K04	Potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania.	K_K03	Potrafi myśleć i działać w sposób przedsiębiorczy.	K_K06	Pracując z zespołem dzieli zadania biorąc pod uwagę słabe i mocne strony każdego z członków zespołu.	K_K05	Rozumie i docenia znaczenie uczciwości intelektualnej w działaniach własnych i innych osób, postępuje etycznie.	K_K01, K_K02, K_K03	
Zna wybrane technologie używane w tworzeniu oprogramowania.	K_W08, K_W10																							
Potrafi rozwijać, uruchamiać i testować oprogramowanie w zadanej technologii. Projektuje rozwiązanie dla postawionego zadania i prezentuje to rozwiązanie. Potrafi ocenić oraz uargumentować zasadność stosowania różnych narzędzi i technik.	K_U03, Inż_U05 K_U03, K_U08, Inż_U05																							
Potrafi zidentyfikować istotny dla siebie fragment kodu w dużym systemie informatycznym i przeanalizować jego działanie.	K_U09, Inż_U01																							
Umie używać popularnych narzędzi do pracy zespołu programistycznego (takich jak Slack, Github).	K_U07, Inż_U02, Inż_U04																							
Umie zrobić rzeczowe „code-review” i nie przyjmuje postawy defensywnej, gdy ktoś zrobi „code-review” jemu.	K_U14, Inż_U02, Inż_U03, Inż_U04																							
Umie korzystać z dokumentacji oraz znajdować w Internecie rozwiązania przy napotkanych przeszkodach technicznych.	K_U13, Inż_U02, Inż_U04																							
Potrafi pracować zespołowo przyjmując w grupie różne role; rozumie konieczność systematycznej pracy nad wszelkimi projektami, które mają długofalowy charakter.	K_K01, K_K04																							
Potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania.	K_K03																							
Potrafi myśleć i działać w sposób przedsiębiorczy.	K_K06																							
Pracując z zespołem dzieli zadania biorąc pod uwagę słabe i mocne strony każdego z członków zespołu.	K_K05																							
Rozumie i docenia znaczenie uczciwości intelektualnej w działaniach własnych i innych osób, postępuje etycznie.	K_K01, K_K02, K_K03																							
16.	<p>Literatura obowiązkowa i zalecana</p> <p>—</p>																							
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>Realizacja zadań w ramach projektu grupowego.</p>																							
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Projekt programistyczny, regularna ewaluacja pracy na spotkaniach (zazwyczaj cotygodniowa).</p>																							
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <table border="1" data-bbox="225 1487 1161 1523"> <tr> <td>pracownia</td> <td>—</td> </tr> </table> <p>Praca własna studenta</p> <table border="1" data-bbox="225 1606 1161 1641"> <tr> <td>Praca nad projektem</td> <td>—</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="225 1724 1161 1792"> <tr> <td>Łączna liczba godzin</td> <td>—</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>—</td> </tr> </table>	pracownia	—	Praca nad projektem	—	Łączna liczba godzin	—	Liczba punktów ECTS	—															
pracownia	—																							
Praca nad projektem	—																							
Łączna liczba godzin	—																							
Liczba punktów ECTS	—																							

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Praktyka zawodowa Student apprenticeship
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu —
6.	Rodzaj przedmiotu/modułu Praktyka zawodowa
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia I stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr nieokreślony
11.	Forma zajęć i liczba godzin praktyka zawodowa — od 75 do 150 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu —
13.	Cele kształcenia dla przedmiotu Podstawowym celem praktyki zawodowej jest przygotowanie studentów do pracy w zawodzie informatyka. Praktykę zawodową należy odbyć na stanowisku pracy uznanym za informatyczne zgodnie z regulaminem praktyk.
14.	Treści programowe Student poznaje środowisko pracy programisty oraz związane z nim zagadnienia; dokładny zakres programowy praktyk zależy od wybranej firmy, realizowanego projektu oraz używanych technologii. Ramowy program praktyk musi uwzględniać: <ul style="list-style-type: none"> • ogólne przygotowanie do pracy na danym stanowisku (zapoznanie z warunkami BHP, zasadami pracy w firmie), • wyznaczone zadania, • wyznaczonego opiekuna po stronie pracodawcy.

15.	<p>Zakładane efekty uczenia się</p> <p>Wiedza</p> <p>Ma podstawową wiedzę dotyczącą prawnych i społecznych aspektów informatyki, w tym odpowiedzialności zawodowej i etycznej, kodeksów etycznych, własności intelektualnej, prywatności i swobód obywatelskich, zna zasady etyki</p> <p>Zna podstawowe pojęcia bezpieczeństwa systemów informatycznych, rozumie ryzyko i odpowiedzialność związane z systemami informatycznymi, rozumie zagrożenia związane z przestępczością elektroniczną</p> <p>Zna podstawowe zasady bezpieczeństwa i higieny pracy w zawodzie informatyka</p> <p>Zna ogólne zasady tworzenia i rozwoju form indywidualnej przedsiębiorczości wykorzystującej wiedzę informatyczną</p> <p>Zna typowe narzędzia stosowane w profesjonalnej pracy informatyka</p> <p>Umiejętności</p> <p>Potrafi rozwijać, uruchamiać i testować oprogramowanie w zadanej technologii</p> <p>Umie używać popularnych narzędzi do pracy zespołu programistycznego (takich jak Slack, Github).</p> <p>Umie korzystać z dokumentacji oraz znajdować w Internecie rozwiązania przy napotkanych przeszkodach technicznych.</p> <p>Potrafi samodzielnie analizować wybrane zagadnienia, dyskutować i prezentować wnioski przed grupą</p> <p>Umie tworzyć oprogramowanie współpracując w grupie</p> <p>Kompetencje społeczne</p> <p>Potrafi pracować zespołowo przyjmując w grupie różne role; rozumie konieczność systematycznej pracy nad wszelkimi projektami, które mają długofalowy charakter</p> <p>Rozumie konieczność ciągłego doskonalenia kompetencji zawodowych spowodowaną szybkimi zmianami technologicznymi</p> <p>Potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania</p> <p>Potrafi myśleć i działać w sposób przedsiębiorczy</p> <p>Ma świadomość ważności i rozumie pozatechniczne aspekty i skutki działalności inżynierskiej, w tym jej wpływu na środowisko, i związanej z tym odpowiedzialności za podejmowane decyzje</p>	<p>K_W10</p> <p>K_W10, Inż_W01</p> <p>K_W10, Inż_W02</p> <p>K_W10, Inż_W01, Inż_W02</p> <p>Inż_W01</p> <p>K_U03, Inż_U05</p> <p>K_U07, K_U14, Inż_U02, Inż_U04</p> <p>K_U07, K_U13, Inż_U02, Inż_U04</p> <p>K_K03</p> <p>K_U14, Inż_U03</p> <p>K_K01, K_K04</p> <p>K_K01</p> <p>K_K03</p> <p>K_K06</p> <p>K_K02</p>						
16.	<p>Literatura obowiązkowa i zalecana</p> <p>W zależności od zadań wyznaczonych podczas praktyki.</p>							
17.	<p>Metody weryfikacji zakładanych efektów uczenia się</p> <p>Potwierdzenie przez firmę informatyczną odbytych praktyk, raport studenta z odbytych praktyk</p>							
18.	<p>Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu</p> <p>Podmiot, w którym student realizuje praktykę wystawia potwierdzenie odbycia praktyk zawierający zakres obowiązków praktykanta oraz używane w procesie technologie. Praktykant, ze swojej strony przygotowuje raport opisujący wykonane zadania. Praktyka zawodowa jest zaliczana na podstawie tych dwóch dokumentów przez uczelnianego opiekuna praktyk. W razie wątpliwości, uczelniany opiekun praktyk może zwrócić się o dodatkowe wyjaśnienia do przedstawicieli firmy lub praktykanta.</p>							
19.	<p>Nakład pracy studenta</p> <p>Zajęcia z udziałem nauczyciela</p> <p>Praca własna studenta</p> <table border="1" data-bbox="231 1836 1420 1881"> <tr> <td>praktyka w wybranym podmiocie</td> <td>od 75 do 150 godz.</td> </tr> </table> <p>Sumarycznie</p> <table border="1" data-bbox="231 1948 1420 2016"> <tr> <td>Łączna liczba godzin</td> <td>75 – 150 godz.</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td>3 – 6</td> </tr> </table>		praktyka w wybranym podmiocie	od 75 do 150 godz.	Łączna liczba godzin	75 – 150 godz.	Liczba punktów ECTS	3 – 6
praktyka w wybranym podmiocie	od 75 do 150 godz.							
Łączna liczba godzin	75 – 150 godz.							
Liczba punktów ECTS	3 – 6							