

1 Obowiązkowe

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Machine Learning Machine Learning
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-ML
6.	Rodzaj przedmiotu/modułu Obowiązkowy
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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 Completed courses: • „Wstęp do informatyki” (Introduction to computer science) or „Algorytmy i struktury danych” (Algorithms and data structures) • Soft requirement of „Sztuczna Inteligencja” (Artificial intelligence) and „Rachunek Prawdopodobieństwa i Statystyka” (Probability theory and statistics). Required Competences: • Python programming, good to know numpy and scipy numerical libraries
13.	Cele kształcenia dla przedmiotu Presentation of fundamentals of machine learning. Presentation of main machine learning techniques: naive Bayes classification, decision trees, and linear regression. Demonstration of practical application of machine learning to real-world problems.
14.	Treści programowe 1. Fundamentals of statistical reasoning. 2. Naive Bayes classifier. 3. Logistic regression. 4. Decision trees. 5. Basics of neural networks. 6. Bias-variance tradeoff, model regularization and ensembling. 7. Boost classifiers, gradient boosting. 8. PAC learning theory. 9. Matrix factorization, recommender systems.

15.	Zakładane efekty uczenia się Knowledge <table border="1"> <tr> <td data-bbox="236 197 1161 264">knows the difference between supervised learning, unsupervised learning, reinforcement learning.</td><td data-bbox="1177 197 1428 264">K_W01</td></tr> <tr> <td data-bbox="236 264 1161 331">knows application domains of machine learning problems (regression, classification, clustering, frequent itemset mining, recommendations)</td><td data-bbox="1177 264 1428 331">K_W01, K_W04</td></tr> <tr> <td data-bbox="236 331 1161 398">knows and understands the probabilistic formulation of machine learning models</td><td data-bbox="1177 331 1428 398">K_W04</td></tr> <tr> <td data-bbox="236 398 1161 465">knows the details of classical machine learning techniques: naive Bayes classifier, linear regression, decision trees</td><td data-bbox="1177 398 1428 465">K_W01, K_W04</td></tr> <tr> <td data-bbox="236 465 1161 577">understands the theoretical (PAC) and practical requirements for success of machine learning projects (understands model bias and variance, knows the importance of proper training, development and test set creation, how to properly measure model performance)</td><td data-bbox="1177 465 1428 577">K_W01, K_W04</td></tr> </table> Skills <table border="1"> <tr> <td data-bbox="236 667 1161 701">can design and execute a machine learning project</td><td data-bbox="1177 667 1428 701">K_U04, K_U05</td></tr> <tr> <td data-bbox="236 701 1161 734">can implement machine learning algorithms</td><td data-bbox="1177 701 1428 734">K_U01, K_U05</td></tr> <tr> <td data-bbox="236 734 1161 857">can choose machine learning techniques suitable for the given problem and appropriate given the nature of the problem, the availability and quality of data, and run-time and training time requirements (costs, computational complexity, need for understandability)</td><td data-bbox="1177 734 1428 857">K_U01, K_U02, K_U03</td></tr> <tr> <td data-bbox="236 857 1161 913">can read and write algorithm description and technical documentation in English</td><td data-bbox="1177 857 1428 913">K_U09</td></tr> </table> Social competences <table border="1"> <tr> <td data-bbox="236 1003 1161 1081">knows the importance of machine learning and artificial intelligence methods for the functioning of society, understands associated opportunities and dangers</td><td data-bbox="1177 1003 1428 1081">K_K02</td></tr> <tr> <td data-bbox="236 1081 1161 1160">shows independent thinking and actions when solving machine learning problems, is aware of the possibility of making mistakes by himself and others, shows criticism of the obtained results</td><td data-bbox="1177 1081 1428 1160">K_K05</td></tr> </table>	knows the difference between supervised learning, unsupervised learning, reinforcement learning.	K_W01	knows application domains of machine learning problems (regression, classification, clustering, frequent itemset mining, recommendations)	K_W01, K_W04	knows and understands the probabilistic formulation of machine learning models	K_W04	knows the details of classical machine learning techniques: naive Bayes classifier, linear regression, decision trees	K_W01, K_W04	understands the theoretical (PAC) and practical requirements for success of machine learning projects (understands model bias and variance, knows the importance of proper training, development and test set creation, how to properly measure model performance)	K_W01, K_W04	can design and execute a machine learning project	K_U04, K_U05	can implement machine learning algorithms	K_U01, K_U05	can choose machine learning techniques suitable for the given problem and appropriate given the nature of the problem, the availability and quality of data, and run-time and training time requirements (costs, computational complexity, need for understandability)	K_U01, K_U02, K_U03	can read and write algorithm description and technical documentation in English	K_U09	knows the importance of machine learning and artificial intelligence methods for the functioning of society, understands associated opportunities and dangers	K_K02	shows independent thinking and actions when solving machine learning problems, is aware of the possibility of making mistakes by himself and others, shows criticism of the obtained results	K_K05
knows the difference between supervised learning, unsupervised learning, reinforcement learning.	K_W01																						
knows application domains of machine learning problems (regression, classification, clustering, frequent itemset mining, recommendations)	K_W01, K_W04																						
knows and understands the probabilistic formulation of machine learning models	K_W04																						
knows the details of classical machine learning techniques: naive Bayes classifier, linear regression, decision trees	K_W01, K_W04																						
understands the theoretical (PAC) and practical requirements for success of machine learning projects (understands model bias and variance, knows the importance of proper training, development and test set creation, how to properly measure model performance)	K_W01, K_W04																						
can design and execute a machine learning project	K_U04, K_U05																						
can implement machine learning algorithms	K_U01, K_U05																						
can choose machine learning techniques suitable for the given problem and appropriate given the nature of the problem, the availability and quality of data, and run-time and training time requirements (costs, computational complexity, need for understandability)	K_U01, K_U02, K_U03																						
can read and write algorithm description and technical documentation in English	K_U09																						
knows the importance of machine learning and artificial intelligence methods for the functioning of society, understands associated opportunities and dangers	K_K02																						
shows independent thinking and actions when solving machine learning problems, is aware of the possibility of making mistakes by himself and others, shows criticism of the obtained results	K_K05																						
16.	Literatura obowiązkowa i zalecana 1. Kevin Murphy, Machine Learning: A Probabilistic Perspective 2. Christopher Bishop, Pattern Analysis and Machine Learning Supplementary reading: • Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning • Leo Breiman: Classification and Regression Trees • online materials for Stanford course CS229.																						
17.	Metody weryfikacji zakładanych efektów uczenia się final exam, lab and written assignments, final project																						
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Lecture final exam Lab assignments The student must obtain at least 60 of points for assignments (about 6 sets per semester) and at least 60 points for the final project (final month of the course). Typical projects deal with implementing a recent research paper, applying methods from the course to new data or research projects settled individually with the instructor.																						

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	lecture	30 h
	classes	30 h
	Praca własna studenta	
	exam	2 h
	work on the project	30 h
	preparation of a report for the final project	5 h
	studying for the final exam	10 h
	preparation for lab assignments	30 h
	preparation for written assignments	25 h
	studying for the lecture and reading literature	10 h
	Sumarycznie	
	Łączna liczba godzin	172 h
	Liczba punktów ECTS	7

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Numerical Optimization Numerical Optimization
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-NumOpt
6.	Rodzaj przedmiotu/modułu Obowiązkowy
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 30 godzin, ćwiczenia – 30 godzin, pracownia – 15 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Prerequisites: • Linear algebra • Multivariate calculus Competences: • Programming skills; Python or Julia is recommended
13.	Cele przedmiotu The aim of this course is to present a detailed survey of optimization methods from both a computational and theoretical perspective. These methods have applications in data sciences, such as machine learning, model fitting, and image processing.
14.	Treści programowe 1. Introduction to optimization 2. Linear programming 3. Convex sets and functions 4. Unconstrained optimization, quadratic optimization 5. Conjugate direction methods 6. Quasi-Newton methods 7. Acceleration, momentum and Nesterov acceleration 8. Constrained optimization, canonical forms and duality 9. Barrier (interior point) methods 10. Proximal methods 11. Alternating direction method of multipliers (ADMM) 12. Subgradient methods 13. Stochastic methods

15.	Zakładane efekty uczenia się	
	Knowledge	
	knows the notion of constraint and unconstraint optimization problems, their canonical forms and duality	K_W01, K_W03
	knows construction of linear programming problems	K_W03
	knows the construction and algorithms used in convex optimization problems	K_W03
	knows the well-known iterative methods (gradient and subgradient, quasi-Newton) for solving convex minimization problems	K_W03
	knows the issue of rate of convergence and knows the Nesterov acceleration methods	K_W03
	knows the interior point methods	K_W03
	Skills	
	can solve simple linear programs using Simplex method and can use the well known software, like CPLEX or Gurobi, to solve more complex ones	K_U04, K_U06, K_U09
	can justify choice of optimization algorithm	K_U04, K_U09
	can justify optimality of a solution	K_U04, K_U09
	can implement first-order steepest descent method	K_U04
	is able to assess the performance of some optimization algorithms	K_U04
	is able to use and implement first-order iterative optimization algorithm related to convex minimization problem.	K_U04
16.	Literatura obowiązkowa i zalecana	
	1. David G. Luenberger, Yinyu Ye, Linear and Nonlinear Programming, Springer 2008. 2. Stephen Boyd, Lieven Vandenberghe, Convex Optimization, Cambridge University Press 2004. 3. Yurii Nesterov, Introductory lectures on convex optimization, Springer 2004. 4. Jorge Nocedal, Stephen J. Wright, Numerical Optimization, Springer 1999.	
17.	Metody weryfikacji zakładanych efektów kształcenia	
	written exam, presentation of problems' solutions (problem solving classes), presentation of source code with solutions of programming assignments (laboratories)	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	1. Exercises: There will be points for presenting solutions of problems. To pass, the sum of 50 of total points must be collected. 2. Laboratories: There will be 6 programming assignments. To pass, the sum of 50 of total points must be collected. 3. Written exam: There will be several problems to solve, student must get the required minimum number of them.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	lecture	30 godz.
	classes+labs	45 godz.
	Praca własna studenta	
	laboratories and problem solving classes	65 h
	self-study	40
	preparing for the exam	20 h
	exam	3 h
	Summary	
	Łączna liczba godzin	203 h
	Liczba punktów ECTS	8

2 Kluczowe przedmioty wybieralne

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Algorytmy ewolucyjne Evolutionary Algorithms
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DZ6AE
6.	Rodzaj przedmiotu/modułu Kluczowy przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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: • znajomość podstaw algebry liniowej • znajomość podstaw rachunku prawdopodobieństwa i statystyki • umiejętność programowania pozwalająca na implementację omawianych algorytmów i przeprowadzanie eksperymentów
13.	Cele przedmiotu Celem przedmiotu jest zapoznanie studentów z algorytmami ewolucyjnymi, a także, bardziej ogólnie, z heurystycznym podejściem do rozwiązywania problemów optymalizacji, zarówno klasycznych problemów optymalizacji, jak i problemów optymalizacji wielomodalnej, wielokryterialnej i dynamicznej.
14.	Treści programowe 1. Wprowadzenie do algorytmów ewolucyjnych. 2. Podstawowe algorytmy ewolucyjne: algorytmy genetyczne, strategie ewolucyjne, programowanie genetyczne, programowanie ewolucyjne. 3. Zaawansowane algorytmy ewolucyjne odkrywające wiedzę o problemie optymalizacji. 4. Równoległe i rozproszone algorytmy ewolucyjne. 5. Algorytmy optymalizacji wielomodalnej. 6. Algorytmy optymalizacji wielokryterialnej. 7. Algorytmy optymalizacji dynamicznej. 8. Wybrane zastosowania algorytmów ewolucyjnych.

15.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>rozumie wybrane podejścia heurystycznych algorytmów optymalizacji</td><td>K_W03, K_W08</td></tr> <tr> <td>rozumie problemy związane z heurystycznymi algorytmami optymalizacji, w tym algorytmami ewolucyjnymi, uwzględniając ich niedokładność i niedeterministyczność</td><td>K_W03, K_W08</td></tr> <tr> <td>posiada przeglądową wiedzę o różnych rodzajach problemów optymalizacji, m.in. optymalizacji wielomodalnej, optymalizacji wielokryterialnej, optymalizacji dynamicznej</td><td>K_W03, K_W08</td></tr> <tr> <td>zna wybrane algorytmy ewolucyjne, zarówno do optymalizacji klasycznej, jak i optymalizacji wielokryterialnej i dynamicznej</td><td>K_W07, K_W08</td></tr> </table> Umiejętności <table> <tr> <td>umie formułować zagadnienia praktyczne jako zadania optymalizacji</td><td>K_U03</td></tr> <tr> <td>umie stosować i modyfikować algorytmy ewolucyjne</td><td>K_U04, K_U08</td></tr> <tr> <td>umie stosować algorytmy optymalizacji wielokryterialnej i dynamicznej</td><td>K_U08, K_U12</td></tr> <tr> <td>umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy</td><td>K_U08, K_U12</td></tr> </table> Kompetencje społeczne <table> <tr> <td>rozumie znaczenie algorytmów optymalizacji dla funkcjonowania współczesnego społeczeństwa</td><td>K_K02</td></tr> </table>	rozumie wybrane podejścia heurystycznych algorytmów optymalizacji	K_W03, K_W08	rozumie problemy związane z heurystycznymi algorytmami optymalizacji, w tym algorytmami ewolucyjnymi, uwzględniając ich niedokładność i niedeterministyczność	K_W03, K_W08	posiada przeglądową wiedzę o różnych rodzajach problemów optymalizacji, m.in. optymalizacji wielomodalnej, optymalizacji wielokryterialnej, optymalizacji dynamicznej	K_W03, K_W08	zna wybrane algorytmy ewolucyjne, zarówno do optymalizacji klasycznej, jak i optymalizacji wielokryterialnej i dynamicznej	K_W07, K_W08	umie formułować zagadnienia praktyczne jako zadania optymalizacji	K_U03	umie stosować i modyfikować algorytmy ewolucyjne	K_U04, K_U08	umie stosować algorytmy optymalizacji wielokryterialnej i dynamicznej	K_U08, K_U12	umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy	K_U08, K_U12	rozumie znaczenie algorytmów optymalizacji dla funkcjonowania współczesnego społeczeństwa	K_K02
rozumie wybrane podejścia heurystycznych algorytmów optymalizacji	K_W03, K_W08																		
rozumie problemy związane z heurystycznymi algorytmami optymalizacji, w tym algorytmami ewolucyjnymi, uwzględniając ich niedokładność i niedeterministyczność	K_W03, K_W08																		
posiada przeglądową wiedzę o różnych rodzajach problemów optymalizacji, m.in. optymalizacji wielomodalnej, optymalizacji wielokryterialnej, optymalizacji dynamicznej	K_W03, K_W08																		
zna wybrane algorytmy ewolucyjne, zarówno do optymalizacji klasycznej, jak i optymalizacji wielokryterialnej i dynamicznej	K_W07, K_W08																		
umie formułować zagadnienia praktyczne jako zadania optymalizacji	K_U03																		
umie stosować i modyfikować algorytmy ewolucyjne	K_U04, K_U08																		
umie stosować algorytmy optymalizacji wielokryterialnej i dynamicznej	K_U08, K_U12																		
umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy	K_U08, K_U12																		
rozumie znaczenie algorytmów optymalizacji dla funkcjonowania współczesnego społeczeństwa	K_K02																		
16.	Literatura obowiązkowa i zalecana wybrane artykuły naukowe z czasopism i konferencji z dziedziny algorytmów ewolucyjnych, inteligencji obliczeniowej i uczenia maszynowego																		
17.	Metody weryfikacji zakładanych efektów kształcenia prezentacja, dyskusja																		
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Zaliczenie pracowni wymaga zdobycia określonej w regulaminie przedmiotu liczby punktów za zadania, projekt i ewentualne dodatkowe aktywności (minireferaty). Zaliczenie wykładu wymaga zdobycia określonej liczby punktów na egzaminie.																		
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>pracownia</td><td>30 godz.</td></tr> <tr> <td>wykład</td><td>30 godz.</td></tr> <tr> <td>udział w egzaminie</td><td>2 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>przygotowywanie się do zajęć (w tym czytanie materiałów dodatkowych)</td><td>20 godz.</td></tr> <tr> <td>rozwiązywanie zadań i realizacja projektu</td><td>60 godz.</td></tr> <tr> <td>przygotowanie do prezentacji projektu i egzaminu</td><td>10 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>152 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.	przygotowywanie się do zajęć (w tym czytanie materiałów dodatkowych)	20 godz.	rozwiązywanie zadań i realizacja projektu	60 godz.	przygotowanie do prezentacji projektu i egzaminu	10 godz.	Łączna liczba godzin	152 godz.	Liczba punktów ECTS	6		
pracownia	30 godz.																		
wykład	30 godz.																		
udział w egzaminie	2 godz.																		
przygotowywanie się do zajęć (w tym czytanie materiałów dodatkowych)	20 godz.																		
rozwiązywanie zadań i realizacja projektu	60 godz.																		
przygotowanie do prezentacji projektu i egzaminu	10 godz.																		
Łączna liczba godzin	152 godz.																		
Liczba punktów ECTS	6																		

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Artificial Intelligence for Games: A Bit of Classics Artificial Intelligence for Games: A Bit of Classics
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DI2AI4GaCla
6.	Rodzaj przedmiotu/modułu Kluczowy przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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 • Artificial Intelligence • proficiency in some (low-level) programming language
13.	Cele przedmiotu Introducing game-specific techniques beyond classic AI methods and showing their advancement in certain problems. Discussing a wide range of algorithms specific to both research and industry, and showing different ways how AI can be applied to games. Last but not least, allowing the students to check their skills in practice by implementing some of the newly-learned algorithms in multiple different game-based environments.
14.	Treści programowe • Minimax and its common optimizations • Unbounded (Best-First) Minimax • B*, Conspiracy Number Search • Proof Number Search, Product Propagation • SSS*, PVS, NegaScout • bitboarding

15.	Zakładane efekty uczenia się Knowledge	
	knows specifics of subdomain of Artificial Intelligence that is focused on games, and different roles that AI can play there	K_W08
	have deep knowledge about practical usage of common AI algorithms, and can choose the best method to solve a given task	K_W03
	knows the recent trends in AI research	K_W07, K_W08
	understands the importance of various AI techniques for the game industry	K_W08
	Skills	
	can implement advanced AI algorithms (e.g. MCTS, RHEA, NegaScout, JPS) and know their most commonly used enhancements of	K_U02, K_U08
	is able to understand and successfully use the methods reported in research publications in the domain	K_U09, K_U12
	can show his project, discuss its result with the group, and write a small publication-like report describing its most important aspects	K_U08, K_U13
	can develop a larger group project requiring research and implementation phases	K_U13
16.	Literatura obowiązkowa i zalecana • Course materials from "Intelligent Search Games" by Prof. Mark Winands, Maastrich University • Publications (provided by the lecturer): AAAI, IJCAI, GECCO, CEC, COG, MCS IJCAI Workshop, etc. • https://chessprogramming.org/	
17.	Metody weryfikacji zakładanych efektów kształcenia presentation of programming tasks, project presentation, written exam	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu To pass the laboratory classes, the student needs to get the required number of points specified in the course program for: • programming tasks during the laboratory classes. • a larger project (preferably in groups), requiring implementation, testing, and documentation of some larger GameAI-related program. Passing laboratory is required to approach written exam testing knowledge on theoretical aspects of presented algorithms. To pass, it is required to get a given number of points. Very good results during the lab classes and approaching bonus assignments may result in exemption from the exam.	
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela	
	lectures	30 h
	classes	30 h
	Praca własna studenta	
	solving programming lab tasks	40 h
	consultations regarding classes and projects	10 h
	writing project report	5 h
	reading provided literature and lecture notes	30 h
	Sumarycznie	
	Łączna liczba godzin	145 h
	Liczba punktów ECTS	6

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 Kluczowy przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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 • General technical culture to learn technology independently. • Knowledge of the English language to allow. – 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 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 <table> <tr> <td>knows the concept of cloud computing</td><td>K_W08</td></tr> <tr> <td>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)</td><td>K_W08</td></tr> <tr> <td>knows the basic services offered in the cloud</td><td>K_W08</td></tr> <tr> <td>knows where to look for knowledge about cloud services</td><td>K_W08</td></tr> </table> Skills <table> <tr> <td>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</td><td>K_U06, K_U08</td></tr> <tr> <td>is able to independently learn cloud technologies</td><td>K_U12</td></tr> <tr> <td>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</td><td>K_U11</td></tr> <tr> <td>is able to adequately address comments and criticism on their proposed solutions</td><td>K_U10</td></tr> <tr> <td>is able to constructively criticize another students' projects</td><td>K_U10</td></tr> <tr> <td>is able to speak English at the B2 level</td><td>K_U09</td></tr> </table>	knows the concept of cloud computing	K_W08	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_W08	knows the basic services offered in the cloud	K_W08	knows where to look for knowledge about cloud services	K_W08	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_U06, K_U08	is able to independently learn cloud technologies	K_U12	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_U11	is able to adequately address comments and criticism on their proposed solutions	K_U10	is able to constructively criticize another students' projects	K_U10	is able to speak English at the B2 level	K_U09
knows the concept of cloud computing	K_W08																				
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_W08																				
knows the basic services offered in the cloud	K_W08																				
knows where to look for knowledge about cloud services	K_W08																				
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_U06, K_U08																				
is able to independently learn cloud technologies	K_U12																				
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_U11																				
is able to adequately address comments and criticism on their proposed solutions	K_U10																				
is able to constructively criticize another students' projects	K_U10																				
is able to speak English at the B2 level	K_U09																				
16.	Literatura obowiązkowa i zalecana Teaching materials offered by cloud service providers - 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 <table> <tr> <td>presentation of own project and discussion on other students' projects</td><td>10 h</td></tr> <tr> <td>presentation of own project idea and discussion on other students' project ideas</td><td>4 h</td></tr> </table> Praca własna studenta <table> <tr> <td>preparation of project proposal and its presentation</td><td>2 h</td></tr> <tr> <td>participation in introductory classes</td><td>1 h</td></tr> <tr> <td>study of selected technologies using the provided teaching materials</td><td>10 h</td></tr> <tr> <td>preparation of the project</td><td>16 h</td></tr> <tr> <td>preparation of the project presentation</td><td>2 h</td></tr> <tr> <td>additional consultations as needed</td><td>2 h</td></tr> </table> Summarycznie <table> <tr> <td>Łączna liczba godzin</td><td>47 h</td></tr> <tr> <td>Liczba punktów ECTS</td><td>2</td></tr> </table>	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	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	Łączna liczba godzin	47 h	Liczba punktów ECTS	2
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																				
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																				
Łączna liczba godzin	47 h																				
Liczba punktów ECTS	2																				

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 Kluczowy przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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: • Algebra Niezbędne kompetencje: • 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 • 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_W03, K_W08
	Knows the concept of LP duality	K_W03, K_W08
	Knows basic algorithms to solve Linear Programms	K_W03, K_W08
	Wiedza	
	Knows the basic theory of Linear Programming	K_W03, K_W08
	Knows the concept of LP duality	K_W03, K_W08
	Knows basic algorithms to solve Linear Programms	K_W03, K_W08
16.	Literatura obowiązkowa i zalecana	
	- 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	
	- Exam - Exercise solving - Writing small computer programms (solving basic LPs with LP solvers)	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	- Collecting enough points for solving exercises and implementation tasks - Passing a final exam	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	lecture	30 h
	classes+labs	30 h
	Praca własna studenta	
	Preparing for the exam	20 h
	Solving theoretical exercises	40 h
	Programming tasks:	20 h
	Sumarycznie	
	Łączna liczba godzin	140 h
	Liczba punktów ECTS	6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs: Obliczenia równoległe na kartach graficznych CUDA (Q2) Parallel Computing with CUDA (Q2)																	
2.	Dyscyplina informatyka																	
3.	Język wykładowy polski																	
4.	Jednostka prowadząca przedmiot Instytut Informatyki																	
5.	Kod przedmiotu/modułu 28-INF-S-ORnKG																	
6.	Rodzaj przedmiotu/modułu Kluczowy przedmiot wybieralny																	
7.	Kierunek studiów (specjalność/specjalizacja) Data Science																	
8.	Poziom studiów studia II stopnia																	
9.	Rok studiów (jeśli obowiązuje) —																	
10.	Semestr zimowy																	
11.	Forma zajęć i liczba godzin wykład – 15 godzin, pracownia – 15 godzin																	
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu • umiejętność biegłego programowania w języku C/C++ • znajomość podstaw grafiki komputerowej • znajomość podstaw architektury systemów komputerowych																	
13.	Cele przedmiotu Głównym celem kursu jest nabycie umiejętności programowania równoległego w technologii CUDA zagadnień nie tylko związanych z grafiką komputerową, a także poznanie architektury współczesnych kart graficznych i ograniczeń, które wpływają na wybór algorytmu i sposób implementacji na karcie graficznej. Dodatkowo poznanie architektury kart graficznych i CUDA API powinno umożliwić zrozumienie i szybszą naukę innych API takich jak np. OpenCL, czy compute shaders z OpenGL.																	
14.	Treści programowe 1. Wstęp do architektury CUDA. 2. Różne rodzaje dostępnej pamięci na GPU i ich szybkość. 3. Architektura CUDA i efektywność: if/atomic/synchr./shared-banks. 4. Warp shuffle, Reduction, Scan. 5. CUDA+OpenGL, przegląd bibliotek, asynchroniczność, strumienie. 6. Cooperative Groups.																	
15.	Zakładane efekty uczenia się Wiedza <table><tr><td>zna architekturę współczesnych kart graficznych (GPU)</td><td>K_W03, K_W08</td></tr><tr><td>zna CUDA API do programowania na GPU</td><td>K_W03, K_W08</td></tr><tr><td>rozumie problemy wydajnościowe i ograniczenia GPU</td><td>K_W03, K_W08</td></tr><tr><td>zna i rozumie klasyczne równoległe operacje typu reduction/scan</td><td>K_W03, K_W08</td></tr></table> Umiejętności <table><tr><td>umie przenieść implementację z CPU na wersję na GPU w technologii CUDA</td><td>K_U06, K_U08</td></tr><tr><td>potrafi stosować narzędzia do debugowania i profilowania na GPU</td><td>K_U06, K_U08</td></tr><tr><td>umie analizować zużycie zasobów i wąskie gardła w programie CUDA</td><td>K_U06, K_U12</td></tr><tr><td>umie zoptymalizować i przyspieszyć działanie naiwnej implementacji CUDA</td><td>K_U06, K_U08</td></tr></table>		zna architekturę współczesnych kart graficznych (GPU)	K_W03, K_W08	zna CUDA API do programowania na GPU	K_W03, K_W08	rozumie problemy wydajnościowe i ograniczenia GPU	K_W03, K_W08	zna i rozumie klasyczne równoległe operacje typu reduction/scan	K_W03, K_W08	umie przenieść implementację z CPU na wersję na GPU w technologii CUDA	K_U06, K_U08	potrafi stosować narzędzia do debugowania i profilowania na GPU	K_U06, K_U08	umie analizować zużycie zasobów i wąskie gardła w programie CUDA	K_U06, K_U12	umie zoptymalizować i przyspieszyć działanie naiwnej implementacji CUDA	K_U06, K_U08
zna architekturę współczesnych kart graficznych (GPU)	K_W03, K_W08																	
zna CUDA API do programowania na GPU	K_W03, K_W08																	
rozumie problemy wydajnościowe i ograniczenia GPU	K_W03, K_W08																	
zna i rozumie klasyczne równoległe operacje typu reduction/scan	K_W03, K_W08																	
umie przenieść implementację z CPU na wersję na GPU w technologii CUDA	K_U06, K_U08																	
potrafi stosować narzędzia do debugowania i profilowania na GPU	K_U06, K_U08																	
umie analizować zużycie zasobów i wąskie gardła w programie CUDA	K_U06, K_U12																	
umie zoptymalizować i przyspieszyć działanie naiwnej implementacji CUDA	K_U06, K_U08																	

16.	Literatura obowiązkowa i zalecana • CUDA Toolkit Documentation: https://docs.nvidia.com/cuda/index.html ("CUDA C Programming Guide", "CUDA C Programming Guide") • David Kirk Wen-mei Hwu - "Programming Massively Parallel Processors", Elsevier (1ed.: 2010, 2ed 2012, 3ed 2016).														
17.	Metody weryfikacji zakładanych efektów kształcenia rozwiązywanie i prezentacja krótkich zadań w postaci programów komputerowych, napisanie i prezentacja programu komputerowego realizującego projekt końcowy														
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Aby zaliczyć kurs należy zrealizować i zaprezentować końcowy projekt programistyczny oraz zdobyć odpowiednią określoną w regulaminie przedmiotu sumę punktów z pracowni i projektu.														
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>pracownia</td><td>15 godz.</td></tr> <tr> <td>wykład</td><td>15 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>przygotowanie do pracowni</td><td>10 godz.</td></tr> <tr> <td>studiowanie tematyki wykładów i literatury</td><td>15 godz.</td></tr> <tr> <td>praca nad projektem</td><td>20 godz.</td></tr> </table> Summarycznie <table> <tr> <td>Łączna liczba godzin</td><td>75 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>3</td></tr> </table>	pracownia	15 godz.	wykład	15 godz.	przygotowanie do pracowni	10 godz.	studiowanie tematyki wykładów i literatury	15 godz.	praca nad projektem	20 godz.	Łączna liczba godzin	75 godz.	Liczba punktów ECTS	3
pracownia	15 godz.														
wykład	15 godz.														
przygotowanie do pracowni	10 godz.														
studiowanie tematyki wykładów i literatury	15 godz.														
praca nad projektem	20 godz.														
Łączna liczba godzin	75 godz.														
Liczba punktów ECTS	3														

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Neural Networks. Theory and Practice. Neural Networks. Theory and Practice.
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DINNThPr
6.	Rodzaj przedmiotu/modułu Kluczowy przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykładu – 30 godzin, pracowni – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Completed Courses: • Introduction to Computer Science (pl. Wstęp do Informatyki) • Linear Algebra • Fundamentals of Programming in Python Prerequisites: • Students should have basic programming skills in Python • Understanding of basic concepts in discrete mathematics, including combinatorics, graph theory, and mathematical logic • Familiarity with fundamental concepts in linear algebra and analytical geometry, including operations on matrices and linear equations.
13.	Cele przedmiotu Goals: • introduce students to the fundamentals of neural networks, including their theoretical foundations and practical applications. • provide hands-on experience with implementing neural networks using the “numpy” and “PyTorch” libraries, as well as the “Lightning” framework. • cover a variety of neural network architectures, including perceptron, multilayer perceptron (MLP), convolutional neural networks (CNN), recurrent neural networks (RNN), and some generative models. • emphasize the importance of good engineering practices, such as testing, debugging, and optimization. • introduce students to a range of neural network applications, including image and video processing, natural language processing, and generative art. • encourage students to think critically about the strengths and limitations of neural networks, and to explore potential ethical considerations related to their use. • provide students with opportunities to work on assignments, homework, and a final project in groups, helping them develop teamwork and collaboration skills while applying their knowledge to real-world problems.

14.

Treści programowe

Program Content

1. Introduction to Neural Networks

• Overview of machine learning and neural networks

• Linear algebra calculus standing behind rtificial neural networks, i.e. multivariate real func-tions.

• Perceptron and multilayer perceptron (MLP) architectures.

2. Convolutional Neural Networks (CNN)

• Motivation for CNNs

• Basics of convolutional operations and pooling

• Common CNN architectures (e.g., LeNet, AlexNet, VGG, ResNet)

• Application of CNNs to image and video processing

3. Recurrent Neural Networks (RNN)

• Motivation for RNNs

• Basics of recurrent operations and backpropagation through time

4. Training Neural Networks

• Cost functions and backpropagation

• Initialization and regularization techniques

• Numerical optimization with stochastic gradient descent

• Hyperparameter tuning and cross-validation

5. Advanced Neural Network Topics

• Transfer learning and fine-tuning

• Object detection and image segmentation tasks

• Generative models (like: Flow-based models or Stable Diffusion)

• Some model interpretability techniques

6. Best Engineering Practices

• Data preprocessing and augmentation

• Batch normalization and other optimization techniques

• Machine Learning Operations (MLOps) and some good practices for ML workflows with Github + ML tracking frameworks (like Weights and Biases or Comet)

15.

Zakładane efekty uczenia się

Knowledge

Knows basics of artificial neural networks	K_W03
Knows common architectures, including perceptron, MLP, CNN, and RNN	K_W03
Knows how to implement and train neural networks using “numpy” and “Py-Torch” libraries	K_W03, K_W04
Knows how to apply neural networks to solve real-world problems, especially in computer vision and natural language processing	K_W05
Knows evaluating neural network models using appropriate metrics	K_W05
Knows interpreting model predictions	K_W05
Knows how to recognize common pitfalls in training and deploying neural net-works	K_W08
Knows best engineering practices for neural network development	K_W07

Skills

Has hands-on skills in implementing and training neural networks using “numpy” and “PyTorch”	K_U01, K_U02, K_U04
Can apply neural networks to solve diverse real-world problems	K_U02, K_U03
Is proficient in evaluating the performance of neural network models using relevant metrics	K_U02, K_U03

Social competences

Collaborating on the final project, where students can apply their knowledge to a real-world problem	K_K01
Communicating findings and interpretations of neural network models	K_K01, K_K02
Developing critical thinking skills in identifying pitfalls and making informed decisions in the development and deployment of neural networks	K_K03
Managing the final project, demonstrating organizational and project manage-ment skills	K_K06

16.	Literatura obowiązkowa i zalecana Literature The course will be based on the following textbook: • Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep Learning. MIT Press. In addition, the following resources may be useful for further reading: • Bishop, C. M. (2006). Pattern recognition and machine learning. Springer. • Zhang, A., Lipton, Z. C., Li, M., Smola, A. J. (2020). Dive into Deep Learning. https://d2l.ai/ • Neural Networks and Deep Learning by Michael Nielsen • Deep Learning with PyTorch by Eli Stevens, Luca Antiga, Thomas Viehmann We will also provide additional materials, including research papers and tutorials, as needed throughout the course.										
17.	Metody weryfikacji zakładanych efektów kształcenia Assessment Methods: • Solution Task Presentation: Presentation and explanation of the solution to a given assignments using neural networks. • Programming Assignments: Writing and presenting a computer program implementing a neural network for a specific task. • Project Presentation: Students will be required to prepare and deliver a presentation on a project related to neural networks. • Written Exam (in needed; see Conditions and Form of Course Completion): A comprehensive written examination covering theoretical concepts and practical applications of neural networks.										
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Conditions and Form of Course Completion: To pass the course, students must accumulate the required number of points for exercises and Programming Assignments, as well as the Project Presentation, which will be held at the end of semester. Points earned for all these activities are considered collectively. While there is a Written Exam, students who achieve a sufficiently high number of points for exercises and the project, as specified in the course regulations, may be exempted from taking the exam. In such cases, the corresponding grade will be transferred from the accumulated points, providing an alternative path to course completion.										
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table border="1"> <tr> <td>lecture</td><td>30 h</td></tr> <tr> <td>labs</td><td>30 h</td></tr> </table> Praca własna studenta <table border="1"> <tr> <td>Student's Independent Work</td><td>90 h</td></tr> </table> Sumarycznie <table border="1"> <tr> <td>Łączna liczba godzin</td><td>150 h</td></tr> <tr> <td>Liczba punktów ECTS</td><td>6</td></tr> </table>	lecture	30 h	labs	30 h	Student's Independent Work	90 h	Łączna liczba godzin	150 h	Liczba punktów ECTS	6
lecture	30 h										
labs	30 h										
Student's Independent Work	90 h										
Łączna liczba godzin	150 h										
Liczba punktów ECTS	6										

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Obliczeniowa teoria uczenia się Computational learning theory
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DI2COLT
6.	Rodzaj przedmiotu/modułu Kluczowy przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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 Prerequisite courses: • Logic for Computer Science (Logika dla informatyków) • Introduction to Computer Science (Wstęp do informatyki) or Algorithms and Data Structures (Algorytmy i struktury danych) • Calculus (Analiza matematyczna) • Discrete Mathematics (Matematyka dyskretna) or Probability theory (Rachunek prawdopodobieństwa) Prerequisite skills: • Mathematical fluency necessary to follow proofs and solve assignments
13.	Cele przedmiotu The course presents theoretical foundation for machine learning methods. The goal of the lecture is to present the theory, which explains why fundamental machine learning algorithms work, and hardness results, which explain why efficient learning in certain scenarios is impossible.
14.	Treści programowe 1. Introduction to PAC. Learnability of particular classes of concepts (DNF formulas, automata). 2. Occam's razor in the PAC context. 3. VC-dimension. Rademacher complexity. 4. Support Vector Machines. 5. Weak and strong learning. Boosting. 6. Online and reinforcement learning. 7. Automata learning.

15.	Zakładane efekty uczenia się	
	Knowledge	
	knows basic machine learning frameworks and their limitations	K_W01, K_W04
	knows connections between various approaches and concepts in machine learning	K_W01, K_W04
	knows constructions of machine learning algorithms	K_W01, K_W04, K_W07
	knows reasons for hardness in machine learning	K_W01, K_W04, K_W07
	Skills	
	can assess complexity of selected machine learning problems	K_U02, K_U08, K_U12
	is able to suggest appropriate methods to facilitate computational learning process	K_U03, K_U12
	can identify hard problems for machine learning algorithms	K_U02, K_U08, K_U12
	Social competences	
	is able to explain strengths and weaknesses of selected machine learning approaches	K_K02, K_K03
	is able to clearly present main ideas in a way tailored towards the audience	K_K01
16.	Literatura obowiązkowa i zalecana • Mehryar Mohri, Foundations of Machine Learning, MIT Press, 2018. • Michael Kearns and Umesh Vazirani. An Introduction to Computational Learning Theory, MIT Press, 1995.	
17.	Metody weryfikacji zakładanych efektów kształcenia written exam, blackboard presentation of solutions, homeworks	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu To pass the course, the student is obliged to pass the classes and the exam. To pass the classes, it is necessary to collect a given number points for presentations and the test. To pass the exam, the student has to solve a given number of assignments.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	lectures	30 godz.
	classes	30 godz.
	Praca własna studenta	
	reading	25 godz.
	problem sets solving	45 godz.
	the exam	2 godz.
	preparation for the exam and solving additional assignments	20 godz.
	Sumarycznie	
	Łączna liczba godzin	152 godz.
	Liczba punktów ECTS	6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Optymalizacja głębokich sieci neuronowych na urządzenia IoT Optimization of deep neural networks for IoT devices
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-OGSNnUIoT
6.	Rodzaj przedmiotu/modułu Kluczowy przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 7 godzin, pracownia – 7 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • (opcjonalnie) zaliczenie przedmiotu Machine Learning (na pierwszych zajęciach przewidziany jest wykład wyrównawczy dla osób, które nie uczęszczały na ML). Niezbędne kompetencje: • znajomość systemu Linux • znajomość języka Python • znajomość języka C++
13.	Cele przedmiotu Celem przedmiotu jest przedstawienie procesu optymalizacji sieci neuronowych pod urządzenia o ograniczonych zasobach obliczeniowych i pamięciowych, jak urządzenia wbudowane.
14.	Treści programowe 1. Algorytmy optymalizujące sieci neuronowe pod względem pamięciowym i obliczeniowym, jak kwantyzacja, pruning, klasteryzacja wag czy destylacja wiedzy. 2. Narzędzia (typu open source) do optymalizacji sieci neuronowych i procesu ich wykonywania pod konkretne urządzenia wbudowane - biblioteki (jak TensorFlow Model Optimization Toolkit) oraz kompilatory (jak TensorFlow Lite czy Apache TVM). 3. Popularne niskopoziomowe techniki optymalizacji najpopularniejszych operacji w sieciach neuronowych. 4. Urządzenia wbudowane z dedykowanymi akceleratorami sieci neuronowych i procesem ich obliczeń. 5. Przykłady głębokich sieci neuronowych projektowanych pod kątem urządzeń wbudowanych.

15.	Zakładane efekty uczenia się	
	Wiedza	
	Zna algorytmy optymalizacji sztucznych sieci neuronowych - quantization, pruning, clustering	K_W03, K_W05, K_W07
	Zna techniki optymalizacji niskopoziomowej sztucznych sieci neuronowych	K_W07, K_W08
16.	Zna podstawy destylacji wiedzy	K_W07, K_W08
	Zna różne biblioteki i narzędzia do uczenia, optymalizacji oraz analizy sztucznych sieci neuronowych, jak TensorFlow, TensorFlow Lite, ONNX oraz TVM	K_W05, K_W07
	Zna różne platformy sprzętowe, na których uruchamiane są sztuczne sieci neuronowe, w szczególności platformy akcelerujące wybrane obliczenia, poznaje również otwartoźródłowe rozwiązania, jak Apache VTA czy CFU Playground	K_W05, K_W07, K_W08
	Umiejętności	
17.	Umie stosować algorytmy kwantyzacji, structured pruningu, unstructured pruningu oraz clusteringu do optymalizacji sieci	K_U02, K_U04
	Umie analizować i porównywać efekty optymalizacji	K_U03, K_U08
	Umie stosować biblioteki i narzędzia do kompilacji sztucznych sieci neuronowych na przykładzie Apache TVM8	K_U06, K_U08
	Umie przeprowadzać konwersje modeli z jednej biblioteki na inną by wykorzystywać różne techniki optymalizacji sieci	K_U06, K_U08, K_U12
18.	Literatura obowiązkowa i zalecana	
	Literatura dostępna online	
	- Aston Zhang, Zachary C. Lipton, Mu Li, Alexander J. Smola. Dive into Deep Learning, https://d2l.ai/ - Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning Book https://www.deeplearningbook.org/	
	Literatura	
19.	- Aurelien Geron, Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow: Concepts, Tools, and Techniques to build Intelligent Systems - Aurelien Geron, Uczenie maszynowe z użyciem Scikit-Learn i TensorFlow: Pojęcia, techniki i narzędzia służące do tworzenia inteligentnych systemów	
	Dokumentacje	
	- TensorFlow documentation, https://www.tensorflow.org/api_docs/python/tf - PyTorch documentation, https://pytorch.org/docs/stable/index.html	
	Metody weryfikacji zakładanych efektów kształcenia	
19.	napisanie i prezentacja programów komputerowych	
	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Do zaliczenia pracowni wymagane jest zdobycie odpowiedniej liczby punktów z ogłoszonych zadań. Dodatkowo, dopuszczalne są dwie nieobecności a każda kolejna zmniejsza ocenę o 1.	
	Dodatkowo, dopuszczalne są dwie nieobecności a każda kolejna zmniejsza ocenę o 1.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	7 godz.
	wykład	7 godz.
19.	Praca własna studenta	
	samodzielne rozwiązywanie zadań pracownianych i projektowych	60 godz.
	Sumarycznie	
	Łączna liczba godzin	74 godz.
19.	Liczba punktów ECTS	3

3 Przedmioty wybieralne

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Algorithmic game theory Algorithmic game theory
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-AlgGameTh
6.	Rodzaj przedmiotu/modułu Przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 45 godzin, ćwiczenia – 15 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Required prior knowledge: • algorithms and data structures Could be helpful, but is not formally required: • some familiarity with computational complexity • basic knowledge of linear programming
13.	Cele przedmiotu The goal of this course is to present various aspects of algorithmic game theory as the research field at the intersection of computational complexity and theory of selfish agents' behaviour.
14.	Treści programowe Noncooperative games: • Equilibria and the complexity of finding them • Lemke-Howson algorithm Cooperative games (coalition forming): • Shapley value • Core • Nucleous Markets, Auctions, Mechanism design: • Exchange markets • Solutions resistant to strategic manipulations

15.	Zakładane efekty uczenia się	
	Knowledge	
	Understands basic concepts of game theory	K_W08
	Knows the computational complexity of finding an equilibrium.	K_W08
	Knows selected concepts from the theory of coalition forming.	K_W08
	Understands the concept of incentive compatibility in the context of mechanism design	K_W08
	Skills	
	Can find an equilibrium for simple two player games	K_U09, K_U12
	Can reason about incentive compatibility of selected mechanisms	K_U09, K_U12
	Social competences	
	Can evaluate (and be critical about) simple decision mechanisms	K_K02
16.	Literatura obowiązkowa i zalecana	
	Nisan, Roughgarden, Tardos, Vazirani. Algorithmic game theory.	
17.	Metody weryfikacji zakładanych efektów kształcenia	
	exam	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	two stage exam (written part and oral part)	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	lectures	45 h
	classes	15 h
	Praca własna studenta	
	literature review	40 h
	preparing for exercise sessions	25 h
	preparing for the exam	20 h
	preparing for a presentation	5 h
	Summary	
	Łączna liczba godzin	150 godz.
	Liczba punktów ECTS	6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Algorytmy funkcjonalne i trwałe struktury danych Functional algorithms and persistent data structures	
2.	Dyscyplina informatyka	
3.	Język wykładowy polski	
4.	Jednostka prowadząca przedmiot Instytut Informatyki	
5.	Kod przedmiotu/modułu 28-INF-S-DZ13AFTSD	
6.	Rodzaj przedmiotu/modułu Przedmiot wybieralny	
7.	Kierunek studiów (specjalność/specjalizacja) Data Science	
8.	Poziom studiów studia II 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 Niezbędne kompetencje • Umiejętność programowania w jakimkolwiek współczesnym języku funkcyjnym: Standard ML-u, Haskellu, OCamlu, F#, Clojure, Scheme itp. • Znajomość podstaw teorii algorytmów i struktur danych (podstawowe struktury danych, takie jak drzewa, kopce itp., postawy analizy złożoności zamortyzowanej). Zrealizowane przedmioty (zalecane, choć nie wymagane) • Algorytmy i struktury danych • Programowanie funkcyjne	
13.	Cele przedmiotu Zapoznanie studentów ze szczególnym działem teorii algorytmów i struktur danych: teorii algorytmów wykonywanych w środowisku, w którym większość struktur danych jest z definicji trwała. Wymaga to zupełnie innego podejścia niż w tradycyjnej teorii ulotnych struktur danych. Mechanizmy odraczania obliczeń dostępne w takich językach jak Standard ML czy OCaml pozwalają na budowanie algorytmów o zadanej złożoności zamortyzowanej działających na trwałych strukturach danych.	
14.	Treści programowe Odpowiadają dokładnie zawartości podręcznika do wykładu.	
15.	Zakładane efekty uczenia się	
	Wiedza	
	Posiada rozszerzoną wiedzę na temat trwałych struktur danych oraz algorytmów operujących na takich strukturach	K_W08
	Zna metody używane przy projektowaniu oraz analizie algorytmów na trwałych strukturach danych	K_W08
	Umiejętności	
	Posiada rozszerzone umiejętności w dziedzinie programowania funkcyjnego Potrafi analizować algorytmy wykorzystujące trwałe struktury danych	K_U08 K_U08

16.	Literatura obowiązkowa i zalecana Podręcznik obowiązkowy - Chris Okasaki, Purely Functional Data Structures, CUP 1999. Literatura dodatkowa - Artykuły z konferencji (np. ICFP) i czasopism (np. J. Functional Programming) opisujące wybrane algorytmy i struktury danych.															
17.	Metody weryfikacji zakładanych efektów kształcenia Ćwiczenia: prezentacja rozwiązań zadań, prezentacja programów komputerowych. Egzamin pisemny.															
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia ćwiczeń należy zdobyć wymaganą liczbę punktów za rozwiązania zadań. Egzamin ma formę pisemną.															
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table><tr><td>wykład</td><td>30 godz.</td></tr><tr><td>ćwiczenia</td><td>30 godz.</td></tr></table> Praca własna studenta <table><tr><td>przygotowanie do egzaminu</td><td>20 godz.</td></tr><tr><td>studiowanie tematyki wykładów i literatury</td><td>30 godz.</td></tr><tr><td>przygotowanie do ćwiczeń</td><td>40 godz.</td></tr></table> Sumarycznie <table><tr><td>Łączna liczba godzin</td><td>150 godz.</td></tr><tr><td>Liczba punktów ECTS</td><td>6</td></tr></table>		wykład	30 godz.	ćwiczenia	30 godz.	przygotowanie do egzaminu	20 godz.	studiowanie tematyki wykładów i literatury	30 godz.	przygotowanie do ćwiczeń	40 godz.	Łączna liczba godzin	150 godz.	Liczba punktów ECTS	6
wykład	30 godz.															
ćwiczenia	30 godz.															
przygotowanie do egzaminu	20 godz.															
studiowanie tematyki wykładów i literatury	30 godz.															
przygotowanie do ćwiczeń	40 godz.															
Łączna liczba godzin	150 godz.															
Liczba punktów ECTS	6															

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Algorytmy kwantowe (Q1) Half-semester course: Quantum algorithms
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-AlgKwantQ2
6.	Rodzaj przedmiotu/modułu Przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład – 15 godzin, ćwiczenia – 15 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • 1/2 kursu Algebry Niezbędne kompetencje: • Podstawy rachunku prawdopodobieństwa • Podstawy algorytmiki • Świadomość istnienia klas złożoności
13.	Cele przedmiotu Zapoznanie studentów z absolutnymi podstawami mechaniki kwantowej i konsekwencjami, jakie ma ona dla informatyki. Wytlumaczenie algorytmu Shora, który funkcjonuje w świadomości informatyków raczej jako hasło niż element wiedzy. Wprowadzenie elementów matematyki (analizy), które są w informatyce kwantowej niezbędne, pojawiają się również w niektórych fragmentach Złożoności Obliczeniowej, ale student informatyki się z nimi raczej nie spotyka.
14.	Treści programowe 1. Polaryzacja liniowa światła. 2. Stan kwantowy i systemy kubitów. 3. Bramki logiczne. 4. Algorytm Grovera. 5. Analiza Fouriera funkcji logicznych, algorytm Simona. 6. Analiza Fouriera nad $\mathbb{Z}_n$. 7. Algorytm Shora. 8. Algorytmy kwantowe i klasy złożoności.

15.	Zakładane efekty uczenia się	
	Wiedza	
	Rozumie, czym różni się konstrukcja algorytmów kwantowych od standardowej algorytmiki	K_W08
	Rozumie, jak decydować, czy algorytmy kwantowe mogą być użyteczne dla zadanego problemu	K_W08
	Zna podstawowe pojęcia mechaniki kwantowej	K_W08
	Umiejętności	
	Umie operować narzędziami analizy fourierowskiej	K_U08, K_U12
	Potrafi przyspieszyć wybrane algorytmy za pomocą wyszukiwania Grovera	K_U08, K_U12
16.	Literatura obowiązkowa i zalecana	
	Ryan O'Donnell Quantum Computation (Materiały do wykładów).	
17.	Metody weryfikacji zakładanych efektów kształcenia	
	egzamin pisemny, ćwiczenia tablicowe	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Do zaliczenia ćwiczeń należy zdobyć wymaganą liczbę punktów za zadania.	
	Egzamin ma formę pisemną. Do zaliczenia go należy zdobyć wymaganą liczbę punktów za zadania.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	ćwiczenia	15 godz.
	wykład	15 godz.
	Praca własna studenta	
	dodatkowe konsultacje w ramach potrzeb	5 godz.
	przygotowanie do egzaminu	10 godz.
	udział w egzaminie	2 godz.
	przygotowywanie się do ćwiczeń (w tym czytanie materiałów dodatkowych)	30 godz.
	Sumarycznie	
	Łączna liczba godzin	77 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Algorytmy probabilistyczne Probabilistic algorithms
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DZ6AP
6.	Rodzaj przedmiotu/modułu Przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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: • Algorytmy i struktury danych • Matematyka dyskretna Niezbędne kompetencje: • Umiejętność analizy złożoności algorytmów • Znajomość pojęć takich jak zmienna losowa, wartość oczekiwana, wariancja.
13.	Cele przedmiotu Celem przedmiotu jest zapoznanie uczestników z podstawowymi technikami używanymi podczas konstruowania i analizy probabilistycznych algorytmów i struktur danych, a także przedstawienie najbardziej klasycznych zastosowań takich rozwiązań.
14.	Treści programowe 1. Podstawowe pojęcia i narzędzia probabilistyki. 2. Zrandomizowane struktury danych (skip listy, treapy). 3. Metoda probabilistyczna, lokalny lemat Lovasza. 4. Łańcuchy Markowa i błędzenie losowe. 5. Zastosowanie losowości w algorytmach grafowych, programowaniu liniowym, geometrii obliczeniowej, algorytmach teorio-liczbowych, itp.

15.	Zakładane efekty uczenia się	
	Wiedza	
	Zna podstawowe techniki dowodzenia poprawności algorytmów zrandomizowanych oraz analizy ich złożoności	K_W03, K_W08
	Zna definicje podstawowych klas złożoności probabilistycznych	K_W08
	Umiejętności	
	Stosując aparat matematyczny potrafi analizować algorytmy zrandomizowane dowodząc ich poprawności oraz szacując ich złożoność	K_U08, K_U12
	Potrafi zrozumieć artykuł naukowy dotyczący algorytmów probabilistycznych	K_U09
	Potrafi przedstawić zastosowania algorytmów zrandomizowanych w realnych systemach komputerowych	K_U08, K_U12
	Potrafi zaprojektować i przeanalizować algorytm zrandomizowany dla wybranych zagadnień	K_U08, K_U12
16.	Literatura obowiązkowa i zalecana	
	• R. Motvani, P. Raghavan: Randomized Algorithms, Cambridge University Press, 1995. • M. Mitzenmacher, U. Eli: Probability and Computing: Randomized Algorithms and Probabilistic Analysis, Cambridge University Press 2005.	
17.	Metody weryfikacji zakładanych efektów kształcenia	
	prezentacja rozwiązania zadania, prezentacja projektu, egzamin pisemny	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Ćwiczenia zaliczane są na podstawie sumy liczby punktów uzyskanych za a) zadeklarowaną umiejętność rozwiązania poszczególnych zadań z otrzymanych list zadań, b) prezentację rozwiązania zadania przy tablicy. Egzamin ma formę pisemną i składa się z dwóch części. Pierwsza sprawdza wiedzę z wykładu za pomocą około 10 krótkich zadań, a druga ma na celu weryfikację umiejętności zastosowania zdobytej wiedzy do projektowania, dowodzenia poprawności oraz analizy złożoności algorytmów probabilistycznych dla podanych 3-4 zadań. Wynik egzaminu jest ustalany na podstawie sumy uzyskanych punktów za poszczególne rozwiązania.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	ćwiczenia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
	udział w egzaminie	5 godz.
	przygotowanie do egzaminu	30 godz.
	samodzielne przygotowanie projektu	15 godz.
	przygotowywanie się do ćwiczeń (w tym czytanie materiałów dodatkowych)	45 godz.
	Sumarycznie	
	Łączna liczba godzin	155 godz.
	Liczba punktów ECTS	6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Algorytmy rozproszone Distributed Algorithms		
2.	Dyscyplina informatyka		
3.	Język wykładowy polski		
4.	Jednostka prowadząca przedmiot Instytut Informatyki		
5.	Kod przedmiotu/modułu 28-INF-S-DZ8AR		
6.	Rodzaj przedmiotu/modułu Przedmiot wybieralny		
7.	Kierunek studiów (specjalność/specjalizacja) Data Science		
8.	Poziom studiów studia II 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: Algorytmy i struktury danych		
13.	Cele przedmiotu Wykład ma na celu zaznajomienie studentów z podstawowymi algorytmami i protokołami występującymi w obliczeniach rozproszonych. Rozważane są modele synchroniczne i asynchroniczne bez wspólnej pamięci. Główny model komunikacji, to wymiana komunikatów między węzłami sieci. Celem jest również rozwinięcie umiejętności analizy poprawności i złożoności takich algorytmów.		
14.	Treści programowe 1. Synchroniczne i asynchroniczne modele obliczeń rozproszonych. 2. Rozproszone algorytmy wykrywania zakleszczenia, wyboru lidera i uzgadniania wspólnej decyzji. 3. Podstawowe algorytmy grafowe w protokołach rozproszonych: algorytmy wyznaczania drzewa rozpinającego i przeszukiwania grafu. 4. Odporność protokołów rozproszonych na uszkodzenia elementów sieci. 5. Przydział zasobów w sieciach.		
15.	Zakładane efekty uczenia się Umiejętności <table border="1"> <tr> <td>Potrafi zaadaptować poznane algorytmy rozproszone do zmodyfikowanych wymagań. Potrafi analizować poprawność i złożoność wybranych algorytmów rozproszonych</td><td>K_U06, K_U08, K_U12 K_U08, K_U12</td></tr> </table>	Potrafi zaadaptować poznane algorytmy rozproszone do zmodyfikowanych wymagań. Potrafi analizować poprawność i złożoność wybranych algorytmów rozproszonych	K_U06, K_U08, K_U12 K_U08, K_U12
Potrafi zaadaptować poznane algorytmy rozproszone do zmodyfikowanych wymagań. Potrafi analizować poprawność i złożoność wybranych algorytmów rozproszonych	K_U06, K_U08, K_U12 K_U08, K_U12		
16.	Literatura obowiązkowa i zalecana 1. N.A. Lynch, Distributed Algorithms, Morgan-Kaufmann Publishers, 1996. 2. G. Tel, Introduction to Distributed Algorithms, Cambridge University Press, 1994. 3. H. Attiya and J. Welsh, Distributed Computing, Fundamentals, Simulations and Advanced Topics, Wiley-Interscience, 2004. 4. N. Santoro, Design and Analysis of Distributed Algorithms, Wiley-Interscience, 2007 5. W. Fokking, Distributed Algorithms, An Intuitive Approach, The MIT Press, 2013.		

17.	Metody weryfikacji zakładanych efektów kształcenia egzamin pisemny, kolokwium, prezentacja rozwiązania zadania																				
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Ćwiczenia zaliczane są na podstawie sumy liczby punktów uzyskanych za: • zadeklarowaną umiejętność rozwiązania poszczególnych zadań z otrzymanych list zadań; • prezentację rozwiązania zadania przy tablicy; • pisemne rozwiązanie zadań w trakcie jednego kolokwium. Egzamin ma formę pisemną i sprawdza wiedzę i umiejętności projektowania, dowodzenia poprawności oraz analizy złożoności algorytmów rozproszonych dla podanych 3-4 zadań. Wynik egzaminu jest ustalany na podstawie sumy uzyskanych punktów za poszczególne rozwiązania.																				
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>ćwiczenia</td><td>30 godz.</td></tr> <tr> <td>wykład</td><td>30 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>przygotowanie do ćwiczeń</td><td>45 godz.</td></tr> <tr> <td>przygotowanie do kolokwium</td><td>6 godz.</td></tr> <tr> <td>udział w egzaminie</td><td>3 godz.</td></tr> <tr> <td>udział w kolokwium</td><td>2 godz.</td></tr> <tr> <td>studiowanie tematyki wykładów i literatury</td><td>22 godz.</td></tr> <tr> <td>przygotowanie do egzaminu</td><td>15 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>153 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>6</td></tr> </table>	ćwiczenia	30 godz.	wykład	30 godz.	przygotowanie do ćwiczeń	45 godz.	przygotowanie do kolokwium	6 godz.	udział w egzaminie	3 godz.	udział w kolokwium	2 godz.	studiowanie tematyki wykładów i literatury	22 godz.	przygotowanie do egzaminu	15 godz.	Łączna liczba godzin	153 godz.	Liczba punktów ECTS	6
ćwiczenia	30 godz.																				
wykład	30 godz.																				
przygotowanie do ćwiczeń	45 godz.																				
przygotowanie do kolokwium	6 godz.																				
udział w egzaminie	3 godz.																				
udział w kolokwium	2 godz.																				
studiowanie tematyki wykładów i literatury	22 godz.																				
przygotowanie do egzaminu	15 godz.																				
Łączna liczba godzin	153 godz.																				
Liczba punktów ECTS	6																				

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kombinatoryka Combinatorics														
2.	Dyscyplina informatyka														
3.	Język wykładowy polski														
4.	Jednostka prowadząca przedmiot Instytut Informatyki														
5.	Kod przedmiotu/modułu 28-INF-S-DZ6KMB														
6.	Rodzaj przedmiotu/modułu Przedmiot wybieralny														
7.	Kierunek studiów (specjalność/specjalizacja) Data Science														
8.	Poziom studiów studia II 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 Zrealizowane przedmioty: • Matematyka dyskretna														
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest zapoznanie studentów z technikami kombinatorycznymi potencjalnie przydatnymi w rozwiązywaniu problemów algorytmicznych.														
14.	Treści programowe • Liczby szczególne • Wzory inwersyjne • Zliczanie istotnie różnych obiektów • Wzór Eulera-Maclaurina i wzór Stirlinga • Teoria Ramseya • Metoda probabilistyczna Erdosa														
15.	Zakładane efekty uczenia się Wiedza <table border="1"> <tr> <td>Zna zagadnienia związane z liczbami szczególnymi</td><td>K_W08</td></tr> <tr> <td>Zna zaawansowane zagadnienia zliczania związane z wzorami inwersyjnymi i teorią Polya</td><td>K_W08</td></tr> <tr> <td>Posiada znajomość teorii Ramsey'a</td><td>K_W08</td></tr> <tr> <td>Zna techniki związane z metodą probabilistyczną</td><td>K_W08</td></tr> </table> Umiejętności <table border="1"> <tr> <td>Umie rozwiązywać proste problemy wymagające użycia liczb szczególnych</td><td>K_U08, K_U12</td></tr> <tr> <td>Umie dla prostych problemów użyć wzorów inwersyjnych i twierdzenia Polya</td><td>K_U08</td></tr> <tr> <td>Umie rozwiązywać problemy używając naiwnej metody probabilistycznej</td><td>K_U08</td></tr> </table>	Zna zagadnienia związane z liczbami szczególnymi	K_W08	Zna zaawansowane zagadnienia zliczania związane z wzorami inwersyjnymi i teorią Polya	K_W08	Posiada znajomość teorii Ramsey'a	K_W08	Zna techniki związane z metodą probabilistyczną	K_W08	Umie rozwiązywać proste problemy wymagające użycia liczb szczególnych	K_U08, K_U12	Umie dla prostych problemów użyć wzorów inwersyjnych i twierdzenia Polya	K_U08	Umie rozwiązywać problemy używając naiwnej metody probabilistycznej	K_U08
Zna zagadnienia związane z liczbami szczególnymi	K_W08														
Zna zaawansowane zagadnienia zliczania związane z wzorami inwersyjnymi i teorią Polya	K_W08														
Posiada znajomość teorii Ramsey'a	K_W08														
Zna techniki związane z metodą probabilistyczną	K_W08														
Umie rozwiązywać proste problemy wymagające użycia liczb szczególnych	K_U08, K_U12														
Umie dla prostych problemów użyć wzorów inwersyjnych i twierdzenia Polya	K_U08														
Umie rozwiązywać problemy używając naiwnej metody probabilistycznej	K_U08														

16.	Literatura obowiązkowa i zalecana • R.L. Graham, D.E. Knuth, O. Patashnik, Matematyka konkretna, PWN 1996. • W.Lipski, W.Marek, Analiza kombinatoryczna, PWN 1988. • Z.Palka, A.Ruciński, Niekonstrukttywne metody matematyki dyskretnej, PWN, Warszawa 1996.																
17.	Metody weryfikacji zakładanych efektów uczenia się Prezentacja rozwiązań zadań, egzamin pisemny																
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Warunkiem zaliczenia jest uzyskanie minimum punktowego za zaprezentowane na tablicy rozwiązania zadań. Warunkiem zaliczenia egzaminu jest zdobycie minimalnej wymaganej liczby punktów z egzaminu końcowego. Szczegółowe warunki zaliczenia są określone w regulaminie przedmiotu.																
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela<table><tr><td>wykład</td><td>30 godz.</td></tr><tr><td>ćwiczenia</td><td>30 godz.</td></tr><tr><td>udział w egzaminie</td><td>2 godz.</td></tr></table> Praca własna studenta<table><tr><td>przygotowanie do egzaminu</td><td>8 godz.</td></tr><tr><td>studiowanie literatury do przedmiotu</td><td>40 godz.</td></tr><tr><td>rozwiązywanie zadań z list</td><td>40 godz.</td></tr></table> Sumarycznie<table><tr><td>Łączna liczba godzin</td><td>150 godz.</td></tr><tr><td>Liczba punktów ECTS</td><td>6</td></tr></table>	wykład	30 godz.	ćwiczenia	30 godz.	udział w egzaminie	2 godz.	przygotowanie do egzaminu	8 godz.	studiowanie literatury do przedmiotu	40 godz.	rozwiązywanie zadań z list	40 godz.	Łączna liczba godzin	150 godz.	Liczba punktów ECTS	6
wykład	30 godz.																
ćwiczenia	30 godz.																
udział w egzaminie	2 godz.																
przygotowanie do egzaminu	8 godz.																
studiowanie literatury do przedmiotu	40 godz.																
rozwiązywanie zadań z list	40 godz.																
Łączna liczba godzin	150 godz.																
Liczba punktów ECTS	6																

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kryptografia Cryptography
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DZ6KRP
6.	Rodzaj przedmiotu/modułu Przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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 • Matematyka dyskretna Zalecane dodatkowo • Algorytmy i struktury danych
13.	Cele kształcenia dla przedmiotu Celem wykładu jest zapoznanie uczestników z nowoczesnymi metodami służącymi do ochrony prywatności danych elektronicznych, autentyfikacji użytkowników systemów komputerowych, zabezpieczaniu przed nieuprawnionymi modyfikacjami danych i innymi tego typu zastosowaniami opartymi na technikach kryptograficznych. Znaczenie tego typu metod ujawnia się szczególnie ostro w epoce powstawania globalnych sieci komputerowych, gdzie systemy operacyjne nie gwarantują już bezpieczeństwa.
14.	Treści programowe • Klasyczne szyfry symetryczne i ich kryptoanaliza. Bezpieczeństwo doskonale i one-time pad • Współczesne szyfry blokowe (DES, IDEA, AES) i ich podstawowe tryby użycia (ECB, CBC, CFB, OFB, CTR) • Kryptoanaliza różnicowa i liniowa • Bezpieczeństwo trybu CBC • Generowanie dużych liczb pierwszych, test Millera-Rabina • Algorytmy faktoryzacji i znajdowania logarytmu dyskretnego • Protokół Diffie-Hellmana. Szyfry asymetryczne RSA i ElGamala. Semantyczne bezpieczeństwo algorytmu ElGamala • Generatory pseudolosowe • Funkcje haszujące • Algorytmy podpisu cyfrowego • Zobowiązania bitowe i dzielenie tajemnic • Protokoły z wiedzą zerową

15.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>zna zagadnienia dotyczące zasad konstrukcji szyfrów blokowych i tryby ich pracy</td><td>K_W08</td></tr> <tr> <td>zna zagadnienia teorioliczne związane z kryptografią asymetryczną</td><td>K_W08</td></tr> <tr> <td>zna kryteria które powinny spełniać generatory pseudolosowe i funkcje skrótu</td><td>K_W08</td></tr> <tr> <td>zna zagadnienia dotyczące podpisów cyfrowych</td><td>K_W08</td></tr> <tr> <td>zna zagadnienia dotyczące dowodów z wiedzą zerową</td><td>K_W08</td></tr> </table> Umiejętności <table> <tr> <td>umie zidentyfikować właściwe sposoby doboru szyfru asymetrycznego wraz z trybem jego działania</td><td>K_U08</td></tr> <tr> <td>umie prawidłowo stosować generatory pseudolosowe, funkcje haszujące i podpisy cyfrowe</td><td>K_U08, K_U12</td></tr> <tr> <td>umie tworzyć proste dowody z wiedzą zerową</td><td>K_U08</td></tr> </table>	zna zagadnienia dotyczące zasad konstrukcji szyfrów blokowych i tryby ich pracy	K_W08	zna zagadnienia teorioliczne związane z kryptografią asymetryczną	K_W08	zna kryteria które powinny spełniać generatory pseudolosowe i funkcje skrótu	K_W08	zna zagadnienia dotyczące podpisów cyfrowych	K_W08	zna zagadnienia dotyczące dowodów z wiedzą zerową	K_W08	umie zidentyfikować właściwe sposoby doboru szyfru asymetrycznego wraz z trybem jego działania	K_U08	umie prawidłowo stosować generatory pseudolosowe, funkcje haszujące i podpisy cyfrowe	K_U08, K_U12	umie tworzyć proste dowody z wiedzą zerową	K_U08
zna zagadnienia dotyczące zasad konstrukcji szyfrów blokowych i tryby ich pracy	K_W08																
zna zagadnienia teorioliczne związane z kryptografią asymetryczną	K_W08																
zna kryteria które powinny spełniać generatory pseudolosowe i funkcje skrótu	K_W08																
zna zagadnienia dotyczące podpisów cyfrowych	K_W08																
zna zagadnienia dotyczące dowodów z wiedzą zerową	K_W08																
umie zidentyfikować właściwe sposoby doboru szyfru asymetrycznego wraz z trybem jego działania	K_U08																
umie prawidłowo stosować generatory pseudolosowe, funkcje haszujące i podpisy cyfrowe	K_U08, K_U12																
umie tworzyć proste dowody z wiedzą zerową	K_U08																
16.	Literatura obowiązkowa i zalecana • Douglas Stinson: Kryptografia. W teorii i praktyce. • Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone: Handbook of applied cryptography. • Neal Koblitz: Algebraiczne aspekty kryptografii.																
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 zaprezentowane przy tablicy zadania. Warunkiem zaliczenia egzaminu jest zdobycie minimalnej wymaganej liczby punktów z egzaminu końcowego. Szczegółowe warunki zaliczenia są określone w regulaminie przedmiotu.																
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>ćwiczenia</td><td>30 godz.</td></tr> <tr> <td>wykład</td><td>30 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>przygotowanie do egzaminu</td><td>8 godz.</td></tr> <tr> <td>studiowanie literatury do przedmiotu</td><td>40 godz.</td></tr> <tr> <td>udział w egzaminie</td><td>2 godz.</td></tr> <tr> <td>rozwiązywanie zadań z list</td><td>40 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>150 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>6</td></tr> </table>	ćwiczenia	30 godz.	wykład	30 godz.	przygotowanie do egzaminu	8 godz.	studiowanie literatury do przedmiotu	40 godz.	udział w egzaminie	2 godz.	rozwiązywanie zadań z list	40 godz.	Łączna liczba godzin	150 godz.	Liczba punktów ECTS	6
ćwiczenia	30 godz.																
wykład	30 godz.																
przygotowanie do egzaminu	8 godz.																
studiowanie literatury do przedmiotu	40 godz.																
udział w egzaminie	2 godz.																
rozwiązywanie zadań z list	40 godz.																
Łączna liczba godzin	150 godz.																
Liczba punktów ECTS	6																

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs języka Elixir Elixir course
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DKElixir
6.	Rodzaj przedmiotu/modułu Przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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 Niezbędne kompetencje: • biegła umiejętność programowania • umiejętność programowania w języku imperatywnym • umiejętność programowania w języku funkcyjnym (Racket też się liczy)
13.	Cele przedmiotu Celem przedmiotu jest zaznajomienie studentów z językiem Elixir, w tym cech które sprawiają, że wyróżnia się na tle innych języków programowania. Chodzi głównie o mechanizmy współbieżności, skalowalności, nadzoru oraz makr. Ponadto zaznajomienie się z najważniejszymi dostępnymi frameworkami i bibliotekami.
14.	Treści programowe 1. Słowa kluczowe, typy, konwencje. 2. Środowisko mix i testy. 3. Maszyna wirtualna erlanga 4. Procesy: współbieżność i komunikacja. 5. Funkcje zaawansowane (protocols, behaviours) 6. OTP (GenServer, Agent, Task, Supervisor) 7. Makra 8. Budowa serwera webowego od podstaw 9. Phoenix 10. Ecto 11. LiveView 12. Absinthe 13. GenStage/Broadway/RabbitMQ

15.	Zakładane efekty uczenia się	
	Wiedza	
	rozumie, i biegle posługuje się językiem ‘Elixir’. Rozumie również, na czym polega specyficzność programowania w nim	K_W08
	posiada przeglądową wiedzę o różnych dostępnych bibliotekach i frameworkach	K_W08
	posiada pogłębioną wiedzę na temat zasad działania i filozofii stojącą za wirtualną maszyną Erlanga	K_W08
	zna podstawowy zestaw narzędzi ‘OTP’ oraz środowisko programistyczne ‘mix’	K_W08
16.	Umiejętności	
	umie modelować różne schematy asynchronicznych aplikacji opartych o mikroserwisy	K_U08
	potrafi implementować własne konstrukcje językowe oraz budować specjalistyczne frameworki i DSL-e w oparciu o makra	K_U08
	umie pisać bezpieczne, skalowalne aplikacje sieciowe	K_U08
	umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy	K_U10
	Literatura obowiązkowa i zalecana	
17.	• Saša Jurić: Elixir in Action • Dave Thomas: Programming Elixir 1.6 • Chris McCord: Metaprogramming Elixir • Svilen Gospodinov: Concurrent Data Processing in Elixir • Bruce Tate and Sophie DeBenedetto: Programming Phoenix LiveView • Bruce Tate, Chris McCord, and Jose Valim: Programming Phoenix • Darin Wilson: Programming Ecto*	
	Metody weryfikacji zakładanych efektów kształcenia	
18.	zadania znajdujące się na listach, projektu końcowy wraz z prezentacją	
	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
19.	Do zaliczenia ćwiczenio-pracowni należy zdobyć wymaganą, podaną w regulaminie przedmiotu liczbę punktów za zadania ćwiczeniowe, pracowniowe. W końcowych tygodniach studenci realizować będą oceniany projekt. Ocena końcowa będzie oceną łączoną z obu form.	
	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	30 godz.
	ćwiczenio-pracownia	30 godz.
	Praca własna studenta	
	przygotowywanie się do ćwiczeń (w tym czytanie materiałów dodatkowych)	30 godz.
	samodzielne rozwiązywanie zadań pracowniowych i projektowych	50 godz.
	Summary	
	Łączna liczba godzin	140 godz.
	Liczba punktów ECTS	5

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs języka Java Java course
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DK6JAV
6.	Rodzaj przedmiotu/modułu Przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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: • wstęp do informatyki Niezbędne kompetencje: • umiejętność programowania strukturalnego i proceduralnego w jakimś języku wyższego poziomu; • pożądana znajomość języka C/C++; • znajomość podstawowych struktur danych; • umiejętność czytania anglojęzycznej dokumentacji.
13.	Cele przedmiotu Celem tego kursu jest nauczanie programowania w jednym z najbardziej popularnych współcześnie języków obiektowych jakim jest Java. Wykład jest ilustrowany programami napisanymi zgodnie z najlepszymi wzorcami, natomiast towarzyszące zajęcia laboratoryjne mają nauczyć dobrych praktyk w programowaniu i projektowaniu obiektowym. Dodatkowo na kursie można się zapoznać z różnymi graficznymi interfejsami GUI obsługiwanymi przez Javę.

14.	Treści programowe	
	1. Podstawowe konstrukcje językowe w Javie. 2. Abstrakcja i hermetyzacja. 3. Dziedziczenie i polimorfizm. 4. Klasy abstrakcyjne, interfejsy. 5. Klasy wewnętrzne, klasy anonimowe. 6. Wyjątki, asercje, dzienniki zdarzeń. 7. Programowanie generyczne. 8. Kolekcje standardowe. 9. Interfejsy graficzne: AWT, Swing, JavaFX; 10. Programowanie współbieżne. 11. Strumienie, serializacja, operacje na plikach. 12. Operacje na łańcuchach znakowych, wyrażenia regularne. 13. Lambdy. 14. Wyliczenia. 15. Refleksja. 16. Adnotacje.	
15.	Zakładane efekty uczenia się	
	Wiedza	
	Student zna standardowe typy danych w języku Java oraz podstawowe konstrukcje programistyczne (instrukcje sterujące, funkcje) wykorzystywane w programowaniu imperatywnym	K_W08
	Student zna proste i złożone typy danych używane w języku Java	K_W08
	Student zna zasady programowania obiektowego w Javie	K_W08
	Student zna praktyczne aspekty złożoności pamięciowej i czasowej programów pisanych w języku wysokiego poziomu	K_W08
	Umiejętności	
	Student potrafi pisać, uruchamiać i testować programy napisane w Javie, używając wybranego środowiska programistycznego	K_U08, K_U12
	Student umie samodzielnie rozwiązywać różne zadania programistyczne korzystając z języka Java	K_U08, K_U12
	Student potrafi wybrać i użyć w swoich programach podstawowych i złożonych struktur danych	K_U08, K_U12
	Student umie podzielić program w Javie na pakiety	K_U08, K_U12
	Kompetencje społeczne	
	Student rozumie znaczenie samodzielnej pracy przy zdobywaniu nowych umiejętności	K_K04
	Student umie zadawać pytania dotyczące rozwiązywania jakiegoś zadania	K_K02
	Student umie opowiedzieć o swoim rozwiązaniu koncentrując się na najbardziej istotnych zagadnieniach	K_K01
16.	Literatura obowiązkowa i zalecana	
	Literatura podstawowa:	
	• Cay S. Horstmann: Java. Podstawy. Wydanie 10. Wydawnictwo HELION, Gliwice 2016.	
	Literatura uzupełniająca:	
	• Cay S. Horstmann: Java 9. Przewodnik doświadczonego programisty. Wydanie 2. Wydawnictwo HELION, Gliwice 2018. • Herbert Schildt: Java. Przewodnik dla początkujących. Wydanie 7. Wydawnictwo HELION, Gliwice 2018.	
	Literatura elektroniczna:	
	• The Java Tutorials (https://docs.oracle.com/javase/tutorial/index.html) • Java Tutorials (https://www.tutorialspoint.com/java_technology_tutorials.htm) • Software Development Entrepreneurship Tutorials (http://tutorials.jenkov.com)	

17.	Metody weryfikacji zakładanych efektów kształcenia Pisanie i prezentacja krótkich programów komputerowych i małych aplikacji (miniprojekty).												
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia przedmiotu należy zdobyć minimalną wymaganą (podaną w regulaminie przedmiotu) liczbę punktów, które uzyskuje się za wykonane w ramach pracowni programy.												
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>pracownia</td><td>30 godz.</td></tr> <tr> <td>wykład</td><td>30 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>przygotowanie do pracowni</td><td>60 godz.</td></tr> <tr> <td>czytanie literatury i dokumentacji</td><td>15 godz.</td></tr> </table> Sumarycznie <table> <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	60 godz.	czytanie literatury i dokumentacji	15 godz.	Łączna liczba godzin	135 godz.	Liczba punktów ECTS	5
pracownia	30 godz.												
wykład	30 godz.												
przygotowanie do pracowni	60 godz.												
czytanie literatury i dokumentacji	15 godz.												
Łączna liczba godzin	135 godz.												
Liczba punktów ECTS	5												

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs: Praktyczne aspekty sieci komputerowych Practical aspects of computer networks
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DKPASK
6.	Rodzaj przedmiotu/modułu Przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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 Ukończenie przedmiotu "Sieci Komputerowe", szkolenia Cisco CCNA lub podobnego w zakresie szkolenia z zakresu sieci, i/lub deklaracja znajomości zagadnień z zakresu materiału w/w kursów.
13.	Cele przedmiotu Celem kursu jest uzupełnienie wiedzy zdobytej na przedmiocie Sieci Komputerowe i/lub CCNA o wiedzę praktyczną z zastosowań wybranych technologii sieciowych z dużym naciskiem na bezpieczeństwo omawianych technologii. Większość omawianych zagadnień będzie do zaimplementowania na specjalnie do tego celu przygotowanych maszynach wirtualnych oraz zajęciowych urządzeniach, żeby jak najlepiej pokazać ich praktyczne zastosowania w firmach i instytucjach. Podczas kursu omawiane będą (między innymi) zagadnienia budowania i administracji sieciami i systemami.

14.	Treści programowe 1. Bezpieczeństwo sieci — zagadnienia takie jak: SIEM, IEEE 802.1x, Radius, VLANy, port-security, MACsec, xyz-snooping, IDS, IPS, itp.) 2. Sieci bezprzewodowe — WiFi 6 (802.11ax), WPA3, PMF, punkty dostępu, zarządzanie, zabezpieczenia, problemy, standardy, bezpieczeństwo 3. Wirtualne sieci prywatne — VPN (IPsec, pptp, OpenVpn, WireGuard, IKEv2, ...) 4. Sieci pamięci masowej — DAS, NAS, SAN, iSCSI, FibreChannel, InfiniBand, vSAN, CEPH, ZFS, wirtualizacja pamięci masowych, bezpieczeństwo w sieciach pamięci masowych 5. Routing i stos TCP/IP w linuxie, Cisco, Juniper, Ubnt, Mikrotik, itp. 6. Protokoły i problemy współczesnych sieci (IPv4 -> IPv6, BGP, MPLS, TCP BBR, SD-LAN, SD-WAN itp.) 7. IPv6 oraz zagadnienia związane z wdrażaniem i integracją z istniejącymi sieciami oraz związane z tym zagadnienia dotyczące bezpieczeństwa sieci 8. Firewall/ACL (iptables, pf, policy routing, ZBF) 9. Zagadnienia związane z wysoką wydajnością oraz rozkładaniem obciążeń — proxy, reverse proxy, loadbalancing, link aggregation, ha, clustering (linux i inne) 10. Wirtualizacja oraz tzw. Cloud Computing — np. VMware, KVM, qemu, XEN, OpenStack, Proxmox 11. Konteneryzacja oraz sieci dla kontenerów — na przykładzie Docker, LXC, Kubernetes 12. Bezpieczeństwo wirtualizacji/cloud computingu/kontenerów 13. Problemy maszyn wirtualnych i kontenerów (i/o, RTC, thin-provisioning, starvation, overbooking, sieci wewnętrzne, ...) 14. Wirtualizacja sieci — wirtualne switchy, routery, karty sieciowe, adresy...																		
15.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>Zna implementacje powyższych mechanizmów we współczesnym Internecie: protokoły TCP, UDP, IP, Ethernet</td><td>K_W08</td></tr> <tr> <td>Zna elementy kryptografii, podstawowe słabości stosowanych protokołów i typy ataków sieciowych</td><td>K_W08</td></tr> <tr> <td>Zna zagadnienia związane z wirtualizacją i konteneryzacją</td><td>K_W08</td></tr> <tr> <td>Zna zaawansowane zagadnienia związane z bezpieczeństwem sieci komputerowych</td><td>K_W08</td></tr> </table> Umiejętności <table> <tr> <td>Jest w stanie zdiagnozować i rozwiązać problemy z brakiem połączenia, połączeniem złej jakości czy też wadliwą konfiguracją routingu</td><td>K_U08</td></tr> <tr> <td>Posługuje się narzędziami szyfrowania danych, potrafi podsłuchiwać nieszyfrowany ruch sieciowy</td><td>K_U08</td></tr> <tr> <td>Potrafi stosować narzędzia do wirtualizacji oraz konteneryzacji</td><td>K_U08</td></tr> <tr> <td>Potrafi zidentyfikować źródła słabej wydajności systemów oraz zaproponować ich rozwiązanie</td><td>K_U08</td></tr> </table> Kompetencje społeczne <table> <tr> <td>Rozumie znaczenie Internetu w funkcjonowaniu współczesnego społeczeństwa, rozumie możliwości i zagrożenia z tym związane</td><td>K_K03</td></tr> </table>	Zna implementacje powyższych mechanizmów we współczesnym Internecie: protokoły TCP, UDP, IP, Ethernet	K_W08	Zna elementy kryptografii, podstawowe słabości stosowanych protokołów i typy ataków sieciowych	K_W08	Zna zagadnienia związane z wirtualizacją i konteneryzacją	K_W08	Zna zaawansowane zagadnienia związane z bezpieczeństwem sieci komputerowych	K_W08	Jest w stanie zdiagnozować i rozwiązać problemy z brakiem połączenia, połączeniem złej jakości czy też wadliwą konfiguracją routingu	K_U08	Posługuje się narzędziami szyfrowania danych, potrafi podsłuchiwać nieszyfrowany ruch sieciowy	K_U08	Potrafi stosować narzędzia do wirtualizacji oraz konteneryzacji	K_U08	Potrafi zidentyfikować źródła słabej wydajności systemów oraz zaproponować ich rozwiązanie	K_U08	Rozumie znaczenie Internetu w funkcjonowaniu współczesnego społeczeństwa, rozumie możliwości i zagrożenia z tym związane	K_K03
Zna implementacje powyższych mechanizmów we współczesnym Internecie: protokoły TCP, UDP, IP, Ethernet	K_W08																		
Zna elementy kryptografii, podstawowe słabości stosowanych protokołów i typy ataków sieciowych	K_W08																		
Zna zagadnienia związane z wirtualizacją i konteneryzacją	K_W08																		
Zna zaawansowane zagadnienia związane z bezpieczeństwem sieci komputerowych	K_W08																		
Jest w stanie zdiagnozować i rozwiązać problemy z brakiem połączenia, połączeniem złej jakości czy też wadliwą konfiguracją routingu	K_U08																		
Posługuje się narzędziami szyfrowania danych, potrafi podsłuchiwać nieszyfrowany ruch sieciowy	K_U08																		
Potrafi stosować narzędzia do wirtualizacji oraz konteneryzacji	K_U08																		
Potrafi zidentyfikować źródła słabej wydajności systemów oraz zaproponować ich rozwiązanie	K_U08																		
Rozumie znaczenie Internetu w funkcjonowaniu współczesnego społeczeństwa, rozumie możliwości i zagrożenia z tym związane	K_K03																		
16.	Literatura obowiązkowa i zalecana • James F. Kurose, Keith W. Ross, Sieci komputerowe. Ujęcie całościowe, Helion, 2018. • Andrew S. Tanenbaum, Sieci komputerowe, Helion, 2012. • Dokumentacja konkretnych narzędzi.																		
17.	Metody weryfikacji zakładanych efektów kształcenia prezentacja rozwiązań zadań																		
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia pracowni należy zdobyć wymaganą, podaną w regulaminie, liczbę punktów.																		

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	30 godz.
	pracownia	30 godz.
	Praca własna studenta	
	przygotowanie raportu/prezentacji	10 godz.
	praca nad projektem	20 godz.
	studiowanie tematyki wykładów i literatury	10 godz.
	przygotowanie do pracowni	20 godz.
	Sumarycznie	
	Łączna liczba godzin	120 godz.
	Liczba punktów ECTS	5

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs programowania gier w silniku Unity Course: Game Development in Unity3D
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DKUnityEng
6.	Rodzaj przedmiotu/modułu Przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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 • podstawy programowania w C (zalecane) • podstawowa znajomość dowolnego programu do modelowania 3D (zalecane) • znajomość języka angielskiego w stopniu umożliwiającym zrozumienie materiałów i tutoriali
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest teoretyczne, a także praktyczne zaznajomienie studentów z procesem projektowania, programowania i testowania gier w silniku Unity3D.
14.	Treści programowe • Projektowanie gier • Podstawy języka C • Pętla gry • Rendering pipeline • Post processing • Podstawy grafiki komputerowej (ścieżki renderingu, rasteryzacja, cieniowanie, śledzenie promieni, macierze)

15.	Zakładane efekty uczenia się Wiedza	
	rozumie proces powstawania i testowania gry zna różne metody monetyzacji gier zna podstawy programowania w języku C# umie dostatkować (juce) grę rozumie pojęcie gamifikacji	K_W08 K_W08, K_W10 K_W08 K_W08 K_W08
	Umiejętności	
	umie zaprojektować, stworzyć oraz przetestować grę używając silnika Unity3D umie zbudować grę na wiele platform (minimalnie: Windows lub macOS i Android lub iOS) umie obsługiwać silnik Unity3D	K_U08, K_U12 K_U08 K_U08
16.	Literatura obowiązkowa i zalecana • https://unity.com/learn • https://www.youtube.com/user/Brackeys	
17.	Metody weryfikacji zakładanych efektów uczenia się napisanie i prezentacja programu komputerowego w postaci gry, prezentacja finalnego projektu	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia pracowni należy zdobyć wymaganą, podaną w regulaminie przedmiotu liczbę punktów za zadania pracowniane (gry) oraz finalny projekt. Punkty za wszystkie wyżej wymienione aktywności liczą się łącznie.	
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela	
	wykład pracownia	30 godz. 30 godz.
	Praca własna studenta	
	dodatkowa praca własna przygotowanie raportu/prezentacji praca nad finalnym projektem	20 godz. 5 godz. 30 godz.
	Sumarycznie	
	Łączna liczba godzin	115 godz.
	Liczba punktów ECTS	5

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs programowania gier w silniku Unreal 5 Course: Game Development in Unreal Engine 5
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DKUnrealEng
6.	Rodzaj przedmiotu/modułu Przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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: • Podstawy programowania w C++
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest teoretyczne, a także praktyczne zaznajomienie studentów z procesem projektowania, programowania i testowania gier w silniku Unreal 5.
14.	Treści programowe • Projektowanie gier • Podstawy języka C • Pętla gry • Rendering pipeline • Post processing • Podstawy grafiki komputerowej (ścieżki renderingu, rasteryzacja, cieniowanie, śledzenie promieni, macierze)

15.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>rozumie proces powstawania i testowania gry</td><td>K_W08</td></tr> <tr> <td>zna różne metody monetyzacji gier</td><td>K_W08, K_W10</td></tr> <tr> <td>zna podstawy programowania w języku C++</td><td>K_W08</td></tr> <tr> <td>umie dostatkować (juce) grę</td><td>K_W08</td></tr> <tr> <td>zna pojęcie gamifikacji</td><td>K_W08</td></tr> </table> Umiejętności <table> <tr> <td>umie zaprojektować, stworzyć oraz przetestować grę używając silnika Unreal 5</td><td>K_U08, K_U12</td></tr> <tr> <td>umie zbudować grę na wiele platform (minimalnie: Windows lub macOS i Android lub iOS)</td><td>K_U08</td></tr> <tr> <td>umie obsługiwać silnik Unreal 5</td><td>K_U08</td></tr> </table>	rozumie proces powstawania i testowania gry	K_W08	zna różne metody monetyzacji gier	K_W08, K_W10	zna podstawy programowania w języku C++	K_W08	umie dostatkować (juce) grę	K_W08	zna pojęcie gamifikacji	K_W08	umie zaprojektować, stworzyć oraz przetestować grę używając silnika Unreal 5	K_U08, K_U12	umie zbudować grę na wiele platform (minimalnie: Windows lub macOS i Android lub iOS)	K_U08	umie obsługiwać silnik Unreal 5	K_U08
rozumie proces powstawania i testowania gry	K_W08																
zna różne metody monetyzacji gier	K_W08, K_W10																
zna podstawy programowania w języku C++	K_W08																
umie dostatkować (juce) grę	K_W08																
zna pojęcie gamifikacji	K_W08																
umie zaprojektować, stworzyć oraz przetestować grę używając silnika Unreal 5	K_U08, K_U12																
umie zbudować grę na wiele platform (minimalnie: Windows lub macOS i Android lub iOS)	K_U08																
umie obsługiwać silnik Unreal 5	K_U08																
16.	Literatura obowiązkowa i zalecana https://www.unrealengine.com/en-US/learn																
17.	Metody weryfikacji zakładanych efektów uczenia się - implementacja i prezentacja programu komputerowego w postaci gry - projekt, realizacja oraz prezentacja finalnego projektu																
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia pracowni należy zdobyć wymaganą, podaną w regulaminie przedmiotu liczbę punktów za zadania pracowniane (gry) oraz finalny projekt. Punkty za wszystkie wyżej wymienione aktywności liczą się łącznie.																
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>wykład</td><td>30 godz.</td></tr> <tr> <td>pracownia</td><td>30 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>dodatkowa praca własna</td><td>20 godz.</td></tr> <tr> <td>przygotowanie raportu/prezentacji</td><td>5 godz.</td></tr> <tr> <td>praca nad finalnym projektem</td><td>30 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>115 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>5</td></tr> </table>	wykład	30 godz.	pracownia	30 godz.	dodatkowa praca własna	20 godz.	przygotowanie raportu/prezentacji	5 godz.	praca nad finalnym projektem	30 godz.	Łączna liczba godzin	115 godz.	Liczba punktów ECTS	5		
wykład	30 godz.																
pracownia	30 godz.																
dodatkowa praca własna	20 godz.																
przygotowanie raportu/prezentacji	5 godz.																
praca nad finalnym projektem	30 godz.																
Łączna liczba godzin	115 godz.																
Liczba punktów ECTS	5																

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 Przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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.	Treści programowe Część 1: DBMS 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 Część 2: System informatyczny 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 * sposoby prezentacji informacji * stronicowanie, sortowanie, filtry * wzorzec specyfikacji 10. Integracja systemów * 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.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>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>K_W08</td></tr> <tr> <td>Zna podstawowe pojęcia nierelacyjnych baz danych</td><td>K_W08</td></tr> <tr> <td>Zna architekturę oraz podstawowe warstwy i komponenty systemu przetwarzającego dane</td><td>K_W08</td></tr> <tr> <td>Zna podstawy DDD i modelowania systemów</td><td>K_W08</td></tr> <tr> <td>Zna podstawowe wzorce i mechanizmy integracji systemów</td><td>K_W08</td></tr> <tr> <td>Zna zasady udostępniania i prezentacji danych</td><td>K_W08</td></tr> <tr> <td>Zna wzorce i podstawy skalowania systemów</td><td>K_W08</td></tr> </table> Umiejętności <table> <tr> <td>Potrafi zaprojektować i zaimplementować relacyjną i nierelacyjną bazę danych</td><td>K_U08, K_U12</td></tr> <tr> <td>Potrafi poprawnie przeprowadzić analizę wymagań dla prostego systemu przetwarzającego dane</td><td>K_U06, K_U08</td></tr> <tr> <td>Potrafi poprawnie zaprojektować prosty system przetwarzający dane</td><td>K_U06, K_U08</td></tr> <tr> <td>Potrafi poprawnie zaimplementować, przetestować i wdrożyć prosty system przetwarzający dane</td><td>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_W08	Zna podstawowe pojęcia nierelacyjnych baz danych	K_W08	Zna architekturę oraz podstawowe warstwy i komponenty systemu przetwarzającego dane	K_W08	Zna podstawy DDD i modelowania systemów	K_W08	Zna podstawowe wzorce i mechanizmy integracji systemów	K_W08	Zna zasady udostępniania i prezentacji danych	K_W08	Zna wzorce i podstawy skalowania systemów	K_W08	Potrafi zaprojektować i zaimplementować relacyjną i nierelacyjną bazę danych	K_U08, K_U12	Potrafi poprawnie przeprowadzić analizę wymagań dla prostego systemu przetwarzającego dane	K_U06, K_U08	Potrafi poprawnie zaprojektować prosty system przetwarzający dane	K_U06, K_U08	Potrafi poprawnie zaimplementować, przetestować i wdrożyć prosty system przetwarzający dane	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_W08																						
Zna podstawowe pojęcia nierelacyjnych baz danych	K_W08																						
Zna architekturę oraz podstawowe warstwy i komponenty systemu przetwarzającego dane	K_W08																						
Zna podstawy DDD i modelowania systemów	K_W08																						
Zna podstawowe wzorce i mechanizmy integracji systemów	K_W08																						
Zna zasady udostępniania i prezentacji danych	K_W08																						
Zna wzorce i podstawy skalowania systemów	K_W08																						
Potrafi zaprojektować i zaimplementować relacyjną i nierelacyjną bazę danych	K_U08, K_U12																						
Potrafi poprawnie przeprowadzić analizę wymagań dla prostego systemu przetwarzającego dane	K_U06, K_U08																						
Potrafi poprawnie zaprojektować prosty system przetwarzający dane	K_U06, K_U08																						
Potrafi poprawnie zaimplementować, przetestować i wdrożyć prosty system przetwarzający dane	K_U08																						
16.	Literatura obowiązkowa i zalecana • 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.	Metody weryfikacji zakładanych efektów kształcenia Prezentacja rozwiązania zadań																						

18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Pracownia: zaliczenie kolejnych list zadań, które prowadzą do kompletnego rozwiązania informatycznego.	
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
	przygotowanie do pracowni	55 godz.
	studiowanie materiału z wykładu	20 godz.
	Sumarycznie	
	Łączna liczba godzin	135 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 Przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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 • Znajomość podstawowych struktur danych: listy, kolejki, drzewa binarne. • Podstawowa umiejętność programowania w Pythonie.
13.	Cele kształcenia dla przedmiotu Celem zajęć jest • 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 • 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	
	zna składnię i semantykę języka Python	K_W08
	zna najpopularniejsze moduły dostępne w Pythonie	K_W08
	Umiejętności	
	potrafi implementować zadania wykorzystując zaawansowane właściwości Pythona, takie jak wbudowane struktury danych, listy składane czy obiekty w nowym stylu	K_U08
	potrafi implementować w Pythonie różne funkcjonalności aplikacji (trwale przechowywanie danych, implementacja interfejsu użytkownika)	K_U08, K_U12
	potrafi zadbać o jakość kodu i zna rekomendacje dot. nazewnictwa i formatowania kodu	K_U08
	potrafi zaprogramować aplikację internetową z wykorzystaniem odpowiedniej platformy (np. django)	K_U08, K_U12
	umie opracować i zaprezentować architekturę rozwiązania informatycznego	K_U08
16.	Literatura obowiązkowa i zalecana • 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.	Metody weryfikacji zakładanych efektów uczenia się • implementacja zadań • realizacja bardziej złożonego projektu	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia pracowni należy zdobyć wskazaną w regulaminie zajęć liczbę punktów za implementację programów oraz za zrealizowanie projektu.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	30 godz.
	pracownia	30 godz.
	Praca własna studenta	
	Implementacja zadań	45 godz.
	realizacja i prezentacja projektu	15 godz.
	Sumarycznie	
	Łączna liczba godzin	120 godz.
	Liczba punktów ECTS	5

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs: zaawansowane techniki w C++ i STL Advanced techniques in C++ and STL
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DKZCPP
6.	Rodzaj przedmiotu/modułu Przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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: • kurs języka C++ • wstęp do informatyki / algorytmy i struktury danych Niezbędne kompetencje: • umiejętność programowania obiektowego w języku C++; • znajomość klasycznych struktur danych wykorzystywanych w algorytmice; • umiejętność czytania anglojęzycznej dokumentacji.
13.	Cele kształcenia dla przedmiotu Celem kursu jest opanowanie przez studentów zaawansowanych technik programowania w nowoczesnej wersji języka C++ oraz maksymalnego wykorzystania możliwości zmodernizowanej biblioteki standardowej STL. Wykład jest ilustrowany programami napisanymi zgodnie z najlepszymi wzorcami, natomiast towarzyszące zajęcia laboratoryjne mają nauczyć dobrych praktyk w programowaniu i projektowaniu obiektowym z wykorzystaniem szablonów.

14.	Treści programowe 1. obsługa błędów i wyjątków 2. łańcuchy i konwersje liczbowe 3. wyrażenia regularne 4. inteligentne wskaźniki 5. zegary i czasomierze 6. obiekty funkcyjne i lambdy 7. kontenery i adaptatory kontenerów 8. iteratory i adaptatory iteratorów 9. iteratory strumieniowe i plikowe 10. algorytmy niemodyfikujące i modyfikujące 11. liczby zespolone i algorytmy numeryczne 12. liczby i rozkłady pseudolosowe 13. współbieżność i synchronizacja wątków 14. metaprogramowanie 15. lokalizacja																				
15.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>zna zasady programowania obiektowego w C++</td><td>K_W08</td></tr> <tr> <td>zna typy szablonowe w języku C++</td><td>K_W08</td></tr> <tr> <td>zna praktyczne aspekty złożoności pamięciowej i czasowej programów pisanych w języku wysokiego poziomu</td><td>K_W08</td></tr> </table> Umiejętności <table> <tr> <td>potrafi pisać, uruchamiać i testować programy napisane w C++, używając wybranego środowiska programistycznego</td><td>K_U08, K_U12</td></tr> <tr> <td>umie samodzielnie rozwiązywać trudne zadania programistyczne korzystając z języka C++ i biblioteki STL</td><td>K_U08, K_U12</td></tr> <tr> <td>potrafi wybrać i użyć w swoich programach struktury danych i algorytmy zdefiniowane w bibliotece standardowej</td><td>K_U08, K_U12</td></tr> <tr> <td>umie podzielić program w C++ na moduły</td><td>K_U08</td></tr> </table> Kompetencje społeczne <table> <tr> <td>Student rozumie znaczenie samodzielnej pracy przy zdobywaniu nowych umiejętności</td><td>K_K05</td></tr> <tr> <td>Student umie zadawać pytania dotyczące rozwiązywania jakiegoś zadania</td><td>K_K01</td></tr> <tr> <td>Student umie opowiedzieć o swoim rozwiązaniu koncentrując się na najbardziej istotnych zagadnieniach</td><td>K_K06</td></tr> </table>	zna zasady programowania obiektowego w C++	K_W08	zna typy szablonowe w języku C++	K_W08	zna praktyczne aspekty złożoności pamięciowej i czasowej programów pisanych w języku wysokiego poziomu	K_W08	potrafi pisać, uruchamiać i testować programy napisane w C++, używając wybranego środowiska programistycznego	K_U08, K_U12	umie samodzielnie rozwiązywać trudne zadania programistyczne korzystając z języka C++ i biblioteki STL	K_U08, K_U12	potrafi wybrać i użyć w swoich programach struktury danych i algorytmy zdefiniowane w bibliotece standardowej	K_U08, K_U12	umie podzielić program w C++ na moduły	K_U08	Student rozumie znaczenie samodzielnej pracy przy zdobywaniu nowych umiejętności	K_K05	Student umie zadawać pytania dotyczące rozwiązywania jakiegoś zadania	K_K01	Student umie opowiedzieć o swoim rozwiązaniu koncentrując się na najbardziej istotnych zagadnieniach	K_K06
zna zasady programowania obiektowego w C++	K_W08																				
zna typy szablonowe w języku C++	K_W08																				
zna praktyczne aspekty złożoności pamięciowej i czasowej programów pisanych w języku wysokiego poziomu	K_W08																				
potrafi pisać, uruchamiać i testować programy napisane w C++, używając wybranego środowiska programistycznego	K_U08, K_U12																				
umie samodzielnie rozwiązywać trudne zadania programistyczne korzystając z języka C++ i biblioteki STL	K_U08, K_U12																				
potrafi wybrać i użyć w swoich programach struktury danych i algorytmy zdefiniowane w bibliotece standardowej	K_U08, K_U12																				
umie podzielić program w C++ na moduły	K_U08																				
Student rozumie znaczenie samodzielnej pracy przy zdobywaniu nowych umiejętności	K_K05																				
Student umie zadawać pytania dotyczące rozwiązywania jakiegoś zadania	K_K01																				
Student umie opowiedzieć o swoim rozwiązaniu koncentrując się na najbardziej istotnych zagadnieniach	K_K06																				
16.	Literatura obowiązkowa i zalecana Literatura podstawowa: • B.Stroustrup: Język C++. Kompendium wiedzy. Wydanie 4. Wydawnictwo Helion, Gliwice 2014. • N.M.Josuttis: C++. Biblioteka standardowa. Podręcznik programisty. Wydanie 2. Wydawnictwo Helion, Gliwice 2014. Literatura uzupełniająca: • J.Grębosz: Opus magnum C++11. Programowanie w języku C++. Tom 1, 2, 3. Wydawnictwo Helion, Gliwice 2018. • J.Galowicz: C++17 STL. Receptury. Wydawnictwo Helion, Gliwice 2018.																				
17.	Metody weryfikacji zakładanych efektów uczenia się Pisanie i prezentacja krótkich programów komputerowych, powiązanych tematycznie z materiałem prezentowanym podczas wykładu.																				
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia przedmiotu należy zdobyć minimalną wymaganą (podaną w regulaminie przedmiotu) liczbę punktów, które uzyskuje się za wykonane w ramach pracowni programy.																				

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	30 godz.
	Sumarycznie	
	Łączna liczba godzin	135 godz.
	Liczba punktów ECTS	5

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Principles of Program Analysis Principles of Program Analysis	
2.	Dyscyplina informatyka	
3.	Język wykładowy angielski	
4.	Jednostka prowadząca przedmiot Instytut Informatyki	
5.	Kod przedmiotu/modułu 28-INF-S-DZ9ANPK	
6.	Rodzaj przedmiotu/modułu Przedmiot wybieralny	
7.	Kierunek studiów (specjalność/specjalizacja) Data Science	
8.	Poziom studiów studia II 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 Required courses: - Logika dla informatyków (Logic for computer scientists) - Metody programowania (Programming methodology) Recommended courses: - Konstrukcja kompilatorów (Compiler construction) - Języki programowania/semantyka języków programowania (Semantics of computer languages) - Programowanie funkcyjne (Functional programming)	
13.	Cele przedmiotu The goal of this course is to introduce students to methods of static program analysis.	
14.	Treści programowe - Data-flow analysis - Control-flow analysis - Abstract interpretation - Type-and-effect systems	
15.	Zakładane efekty uczenia się	
	Knowledge	
	knows the foundations of static program analysis, with its capabilities and limitations	K_W08
	knows terminology and mathematical frameworks used in program analysis	K_W08
	knows the main techniques used in program analysis	K_W08
15.	knows classical analyses	
	Skills	
	can analyze simple properties of programs	K_U08, K_U12
	can implement his/her own analyses	K_U08, K_U12
	can apply program-analysis methods to solve software engineering and security problems	K_U08, K_U12

16.	Literatura obowiązkowa i zalecana Flemming Nielson, Hane Riis Nielson, Chris Hankin. Principles of Program Analysis. Uday P. Khedker, Amitabha Sanyal, Bageshri Karkare. Data Flow Analysis: Theory and Practice. CRC Press, 2009.	
17.	Metody weryfikacji zakładanych efektów kształcenia presentation of problem solutions in class, inividual programming exercises, written exam	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu To pass the course, the student is obliged to pass the classes and the exam. To pass the classes, it is necessary to collect a given number points for presentations and homeworks (considered separately). To pass the exam, the student has to solve a given number of assignments.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	lectures	30 h
	classes	30 h
	Praca własna studenta	
	reading	20 h
	preparation for class	20 h
	exam	3 h
	preparation for exam	15 h
	programming homework	30 h
19.	Summarycznie	
	Łączna liczba godzin	148 h
	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 Przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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 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 - 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.	Zakładane efekty uczenia się	
	Wiedza	
	zna zasady projektowania obiektowego	K_W08
	zna wybrane wzorce projektowe	K_W08
	zna wybrane wzorce architektury aplikacji	K_W08
16.	zna język UML	K_W08
	zna wybrane metodyki wytwarzania aplikacji	K_W08
	zna sposoby testowania aplikacji	K_W08
	Umiejętności	
	potrafi zbudować dokumentację analityczną systemu przy pomocy wybranego narzędzia do projektowania UML	K_U08
17.	potrafi projektować architekturę systemu w zależności od wymagań	K_U08, K_U12
	potrafi implementować poszczególne elementy stosu aplikacyjnego zgodnie z praktyką przemysłową	K_U08, K_U12
	Kompetencje społeczne	
	rozumie konieczność ciągłego doskonalenia się w obliczu zmieniających się technologii	K_K05
18.	Literatura obowiązkowa i zalecana	
	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#.	
19.	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	
20.	Metody weryfikacji zakładanych efektów kształcenia	
	• napisanie i prezentacja programu komputerowego	
	• prezentacja rozwiązania zadania	
	• egzamin pisemny	
21.	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.	
	Egzamin pisemny sprawdzający wiedzę w poszczególnych partiach materiału.	
22.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
23.	udział w egzaminie	2 godz.
	przygotowanie do pracowni	40 godz.
	studiowanie tematyki wykładów i literatury	20 godz.
	przygotowanie do egzaminu	20 godz.
	Sumarycznie	
24.	Łączna liczba godzin	142 godz.
	Liczba punktów ECTS	6

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 Przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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: • 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.	Treści programowe	
	Program:	
	I. Scala syntax	
	• Types • Classes and objects • Traits • Functions and closures • Collections • Case classes and pattern matching • Lambdas • Implicit parameters • Code standards	
	II. Frameworks and libraries	
	• Build tool - [Sbt] • Testing - [ScalaTest] • Database access - [Slick] • Web applications - [Play Framework] • Concurrent and distributed applications [Akka] • Functional Programming - [Cats]	
	III. Glimpse into the future of Scala [Dotty compiler]	
15.	Zakładane efekty uczenia się	
	Knowledge	
	knows the syntax of Scala language	K_W08
	knows basics of object-oriented and functional programming	K_W08
	has overview of standards and frameworks used in the industry	K_W08
	Skills	
	can implement a medium size web-based application (Play Akka)	K_U08
	can write unit-tests (ScalaTest)	K_U08
	can use build tool (Sbt)	K_U08
16.	Literatura obowiązkowa i zalecana	
	• 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.	Metody weryfikacji zakładanych efektów kształcenia	
	Programming exercises and demos.	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	To pass the course, one needs to gain the amount of points that are indicated in the regulations.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	lecture	30 h
	labs	30 h
	Praca własna studenta	
	self study	20 h
	programming	45 h
	Sumarycznie	
	Łączna liczba godzin	125 h
	Liczba punktów ECTS	5

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Scheduling theory Scheduling theory
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-SchedThr
6.	Rodzaj przedmiotu/modułu Przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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 Required prerequisite courses: • None. Recommended courses: • Discrete Mathematics • Algorithms and Data Structures • Approximation Algorithms • Online Algorithms
13.	Cele przedmiotu This course focuses on scheduling theory as a branch of theoretical computer science, and our main goal is to design optimal algorithms – or, if not optimal, then at least provably efficient ones. Throughout the course, we shall introduce the main algorithmic tools used for in the area of scheduling theory, and present select major results in the area which use these tools. In particular, we will focus on optimal polynomial-time algorithms for scheduling problems, approximation algorithms, and online algorithms.
14.	Treści programowe Sample topics: 1. Introduction to scheduling theory, overview of the area, Graham notation. 2. List scheduling, online list scheduling. List scheduling on machines with different speeds. 3. Dynamic programming and its applications in scheduling. 4. Preemptive list scheduling minimizing the sum of weighted completion times. 5. Scheduling on unrelated parallel machines. 6. Online throughput scheduling of unit and equal-size jobs. 7. Semi-online scheduling: known optimum makespan, known sum of processing times.

15.	Zakładane efekty uczenia się	
	Knowledge	
	The student knows the basic concepts of scheduling, and is able to formulate key problems in the area.	K_W08
	The student is knowledgeable with the Graham scheduling notation.	K_W08
	Out of the key problems in the area, the student knows which are NP-complete and what results are known in the approximation and online algorithm setting.	K_W08
	Skills	
	The student knows the basic tools for design of optimal, approximate and online algorithms for scheduling, including dynamic programming, greedy strategies, deterministic rounding of linear programs, and more.	K_U08, K_U12
	The student is able to solve simple exercises utilizing those tools individually.	K_U08, K_U12
16.	Literatura obowiązkowa i zalecana • M. L. Pinedo. Scheduling: Theory, Algorithms, and Systems. Springer, 2008. • Peter Brucker: Scheduling Algorithms. Springer, 2007. • D. P. Williamson, D. B. Shmoys: The Design of Approximation Algorithms. Cambridge University Press, 2012. Available online: https://www.designofapproxalgs.com/ • B. Chen, C. N. Potts, G. J. Woeginger: A review of machine scheduling: Complexity, algorithms and approximability. In D. Z. Du and P. M. Pardalos. Handbook of Combinatorial Optimization, 1998.	
17.	Metody weryfikacji zakładanych efektów kształcenia The lecture will have an oral exam, consisting of presenting a simple and an intermediate topic from the lecture. The student will have some time to prepare his oral presentation, and will then discuss the topic with the lecturer.	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu The exercise sessions will consist of problem solving practice in class, and grades for the exercise sessions will be determined by two larger homework sheets throughout the semester.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	lecture	30 h
	classes	30 h
	Praca własna studenta	
	exam participation	3 h
	Homework preparation	45 h
	additional consultations, if required	1 h
	Preparation for the exam	25 h
	literature study	15 h
	Summary	
	Łączna liczba godzin	149 h
	Liczba punktów ECTS	6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Teoria kategorii w informatyce Category Theory in Computer Science
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DI2CatThe
6.	Rodzaj przedmiotu/modułu Przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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
13.	Cele przedmiotu Celem przedmiotu jest zapoznanie słuchaczy z podstawowymi pojęciami teorii kategorii i związkami tych pojęć z logiką, algebrą i konstrukcjami znanymi z języków programowania.
14.	Treści programowe • Podstawowe pojęcia teorii kategorii: kategorie, funktory, transformacje naturalne, • Konstrukcje w kategoriach (tłumaczenie definicji z języka teorii zbiorów na język teorii kategorii) i konstrukcje na kategoriach (np. produkt kategorii, kategorie strzałkowe i przecinkowe), • Dualność: dwa pojęcia/twierdzenia w cenie jednego, • Konstrukcje uniwersalne: (ko)kończowość, (ko)produkty, (ko)granice, sprzężenia, • Lemat Yoneda, • Domknięte kategorie kartezjańskie jako modele rachunku lambda, • (Ko)rekurencyjne typy danych definiowane jako algebry (ko)końcowe, • Monady i ich algebry, • Wolne monady i semantyka efektów algebraicznych, • Kwantyfikatory jako sprzężenia.

15.	Zakładane efekty uczenia się	
	Wiedza	
	zna najważniejsze pojęcia teorii kategorii	K_W08
	zna interpretacje podstawowych pojęć matematyki przy pomocy własności uniwersalnych	K_W08
	jest świadomy związku teorii kategorii z logiką, algebrą abstrakcyjną i językami programowania	K_W08
	zna sposoby wyrażania semantyki programów komputerowych przy użyciu języka teorii kategorii	K_W08
	Umiejętności	
	umie rozpoznawać pojęcia teorii kategorii w matematyce i programowaniu	K_U08, K_U12
	umie przeprowadzać rozumowania na wysokim poziomie abstrakcji	K_U08, K_U12
	umie wykorzystywać pojęcia teorii kategorii do konstrukcji programów komputerowych	K_U08, K_U12
	umie wykorzystywać narzędzia teorii kategorii do wnioskowania o poprawności algorytmów	K_U08, K_U12
	Kompetencje społeczne	
	umie prezentować abstrakcyjne pojęcia w sposób dostosowany do wiedzy słuchaczy	K_K01, K_K02
16.	Literatura obowiązkowa i zalecana	
	• Steve Awodey, Category Theory (2nd ed.), Oxford Logic Guides, 2010 • Saunders Mac Lane, Categories for the Working Mathematician (2nd ed.), Springer, 1998 • Michael Barr, Charles Wells, Category Theory for Computing Science, Prentice Hall, 1990 • Harold Simmons, An Introduction to Category Theory, Cambridge University Press, 2011 • David Spivak, Category Theory for the Sciences, MIT Press, 2014 • Tom Leinster, Basic Category Theory, Cambridge University Press, 2014	
17.	Metody weryfikacji zakładanych efektów kształcenia	
	egzamin pisemny, prezentacja rozwiązania zadania	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	wykład: egzamin pisemny w formie pracy domowej ćwiczenia: dwie prace domowe i aktywność na zajęciach	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	ćwiczenia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
	prace domowe	20 godz.
	egzamin	5 godz.
	przygotowanie do ćwiczeń	60 godz.
	Sumarycznie	
	Łączna liczba godzin	145 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 Przedmiot wybieralny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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.	Treści programowe • Pojęcia: problem, specyfikacja, algorytm, program komputerowy. Budowa komputera – schemat logiczny. • Elementy programowania w języku niskopoziomym 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.	Zakładane efekty uczenia się Wiedza <table border="1"> <tr> <td>zna i rozumie pojęcie specyfikacji problemu algorytmicznego, zgodności algorytmu ze specyfikacją i sposobu wykonywania programów przez komputer</td><td>K_W08</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_W08</td></tr> </table> Umiejętności <table border="1"> <tr> <td>potrafi formułować precyzyjną specyfikację problemu</td><td>K_U08</td></tr> <tr> <td>konstruuje algorytmy rozwiązujące standardowe problemy sformułowane w postaci specyfikacji</td><td>K_U08</td></tr> <tr> <td>potrafi analizować poprawność i złożoność obliczeniową prostych algorytmów, w tym rekurencyjnych</td><td>K_U08</td></tr> <tr> <td>wykorzystuje podstawowe techniki algorytmiczne, struktury danych i obiekty abstrakcyjne (np. grafy) do rozwiązywania problemów</td><td>K_U08, K_U12</td></tr> <tr> <td>potrafi interpretować i zapisywać elementy składni języków programowania w postaci gramatyk bezkontekstowych</td><td>K_U08</td></tr> <tr> <td>posługuje się cyfrową reprezentacją danych: liczb, obrazów, grafów</td><td>K_U08</td></tr> </table>	zna i rozumie pojęcie specyfikacji problemu algorytmicznego, zgodności algorytmu ze specyfikacją i sposobu wykonywania programów przez komputer	K_W08	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_W08	potrafi formułować precyzyjną specyfikację problemu	K_U08	konstruuje algorytmy rozwiązujące standardowe problemy sformułowane w postaci specyfikacji	K_U08	potrafi analizować poprawność i złożoność obliczeniową prostych algorytmów, w tym rekurencyjnych	K_U08	wykorzystuje podstawowe techniki algorytmiczne, struktury danych i obiekty abstrakcyjne (np. grafy) do rozwiązywania problemów	K_U08, K_U12	potrafi interpretować i zapisywać elementy składni języków programowania w postaci gramatyk bezkontekstowych	K_U08	posługuje się cyfrową reprezentacją danych: liczb, obrazów, grafów	K_U08
zna i rozumie pojęcie specyfikacji problemu algorytmicznego, zgodności algorytmu ze specyfikacją i sposobu wykonywania programów przez komputer	K_W08																				
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_W08																				
potrafi formułować precyzyjną specyfikację problemu	K_U08																				
konstruuje algorytmy rozwiązujące standardowe problemy sformułowane w postaci specyfikacji	K_U08																				
potrafi analizować poprawność i złożoność obliczeniową prostych algorytmów, w tym rekurencyjnych	K_U08																				
wykorzystuje podstawowe techniki algorytmiczne, struktury danych i obiekty abstrakcyjne (np. grafy) do rozwiązywania problemów	K_U08, K_U12																				
potrafi interpretować i zapisywać elementy składni języków programowania w postaci gramatyk bezkontekstowych	K_U08																				
posługuje się cyfrową reprezentacją danych: liczb, obrazów, grafów	K_U08																				
16.	Literatura obowiązkowa i zalecana • 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.	Metody weryfikacji zakładanych efektów uczenia się prezentacja rozwiązania zadania, kolokwium, egzamin pisemny																				

18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu 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.																					
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table><tr><td>ćwiczenia</td><td>30 godz.</td></tr><tr><td>wykład</td><td>30 godz.</td></tr></table> Praca własna studenta <table><tr><td>udział w kolokwiach i egzaminie</td><td>5 godz.</td></tr><tr><td>analiza i weryfikacja notatek z zajęć oraz innych źródeł informacji</td><td>15 godz.</td></tr><tr><td>nabycie umiejętności posługiwania się symulatorem maszyny RAM –</td><td>5 godz.</td></tr><tr><td>przygotowanie do kolokwiów</td><td>15 godz.</td></tr><tr><td>przygotowanie do egzaminu</td><td>20 godz.</td></tr><tr><td>rozwiązywanie zadań z list</td><td>30 godz.</td></tr></table> Sumarycznie <table><tr><td>Łączna liczba godzin</td><td>150 godz.</td></tr><tr><td>Liczba punktów ECTS</td><td>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																					

4 Seminars

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminar: Advanced Data Mining for Complex and Unstructured Data Seminar: Advanced Data Mining for Complex and Unstructured Data
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DSAda
6.	Rodzaj przedmiotu/modułu Seminarium
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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 Necessary competencies: • knowledge of the basics of linear algebra • knowledge of the basics of probability and statistics • knowledge of the basics of data mining or determination to learn them on your own in the first weeks of the semester
13.	Cele przedmiotu The aim of the seminar is to familiarize students with advanced data mining algorithms related to the processing of complex and unstructured data, mainly regarding recommender systems, dimensionality reduction, temporal data mining and decision support systems.
14.	Treści programowe Selected algorithms for mining complex and unstructured data for: • recommender systems, including systems based on a rating matrix and systems based on session records, • dimensionality reduction, especially based on the embedding of point neighborhoods, • processing temporal data, including training stochastic state models, • decision support systems.

15.	Zakładane efekty uczenia się Knowledge	
	understands basic approaches to mining complex and unstructured data	K_W02, K_W08
	understands problems related to mining complex and unstructured data	K_W02, K_W08
	has overview knowledge of various approaches to mining complex and unstructured data	K_W08
	knows selected basic algorithms for mining complex and unstructured data	K_W06, K_W08
	Skills	
	is able to formulate practical issues as tasks for mining complex and unstructured data	K_U06, K_U07
	is able to use and modify basic algorithms for mining complex and unstructured data	K_U08
	is able to present his ideas in a way adapted to the listeners' knowledge (K_U12)	K_U09, K_U10, K_U11
	Social competences	
	understands the importance of mining complex and unstructured data for the functioning of modern society	K_K02
16.	Literatura obowiązkowa i zalecana selected scientific articles from journals and conferences in the field of data mining, computational intelligence and machine learning	
17.	Metody weryfikacji zakładanych efektów kształcenia presentation, discussion	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Passing the seminar requires preparing and presenting a talk on the topic assigned by the instructor.	
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela	
	seminar	30 h
	Praca własna studenta	
	study of literature and additional materials	30 h
	presentation preparation	15 h
	Sumarycznie	
	Łączna liczba godzin	75 h
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminar: Deep Learning in Biomedical Image Analysis Seminar: Deep Learning in Biomedical Image Analysis	
2.	Dyscyplina informatyka	
3.	Język wykładowy angielski	
4.	Jednostka prowadząca przedmiot Instytut Informatyki	
5.	Kod przedmiotu/modułu 28-INF-S-DHSEGI	
6.	Rodzaj przedmiotu/modułu Seminarium	
7.	Kierunek studiów (specjalność/specjalizacja) Data Science	
8.	Poziom studiów studia II 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 Completed courses: - Machine learning - Course on Biomedical Image Analysis for Computer Scientists	
13.	Cele przedmiotu Deep learning is rapidly revolutionising biomedical image image analysis. Yet, unlike fields where deep learning is applied to intuitively understandable data domains like photographs, biomedical images are often acquired under domain-specific assumptions using complex equipment to solve tasks with high equivocation. Specialised deep learning algorithms take these priors into account making such scientific papers hard to understand for a computer scientist lacking the necessary background. In this seminar series, we will explore the most recent scientific literature involving deep learning algorithms used in biomedical image analysis. In a group effort, we will understand the priors and assumptions of biomedical image acquisition and analysis and understand the cutting-edge approaches.	
14.	Treści programowe -	
15.	Zakładane efekty uczenia się Knowledge	
	Knows state-of-the-art methods in biomedical image analysis using deep learning	K_W04, K_W05
	Knows recent trends in biomedical image analysis	K_W07, K_W08
	Skills	
	Can conduct a critical analysis of a given topic	K_U10, K_U12
	Can prepare a clear presentation with slides of a given topic on biomedical image analysis	K_U11
16.	Literatura obowiązkowa i zalecana Selected recent research papers	

17.	Metody weryfikacji zakładanych efektów kształcenia presentation, discussion										
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu To pass the seminar, a student has to give a presentation on the selected topic.										
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>seminar</td><td>30 h</td></tr> </table> Praca własna studenta <table> <tr> <td>preparing the presentation</td><td>15 h</td></tr> <tr> <td>literature study</td><td>30 h</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>75 h</td></tr> <tr> <td>Liczba punktów ECTS</td><td>3</td></tr> </table>	seminar	30 h	preparing the presentation	15 h	literature study	30 h	Łączna liczba godzin	75 h	Liczba punktów ECTS	3
seminar	30 h										
preparing the presentation	15 h										
literature study	30 h										
Łączna liczba godzin	75 h										
Liczba punktów ECTS	3										

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminar: Graph Neural Networks and Applications Seminar: Graph Neural Networks and Applications
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DSGrapNN
6.	Rodzaj przedmiotu/modułu Seminarium
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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 Necessary competencies: • knowledge of the basics of linear algebra • knowledge of the basics of probability and statistics • knowledge of the basics of machine learning and neural networks or determination to learn them on your own in the first weeks of the semester
13.	Cele przedmiotu The aim of the seminar is to familiarize students with the basic algorithms of graph neural networks and their contemporary applications in practice and existing tools.
14.	Treści programowe 1. Models of graph neural networks. 2. Algorithms for learning graph neural networks. 3. Applications of graph neural networks.

15.	Zakładane efekty uczenia się Knowledge	
	understands basic GNN approaches understands problems related to GNN has overview knowledge of various GNN approaches knows selected basic GNN algorithms	K_W02, K_W08 K_W02, K_W08 K_W08 K_W06, K_W08
	Skills	
	is able to formulate practical issues as tasks for GNN is able to use and modify basic GNN algorithms is able to present his ideas in a way adapted to the listeners' knowledge	K_U06, K_U07 K_U08 K_U09, K_U10, K_U11
	Social competences	
	understands the importance of GNN learning algorithms for the functioning of modern society	K_K02
16.	Literatura obowiązkowa i zalecana selected scientific articles from journals and conferences in the field of data mining, computational intelligence and machine learning	
17.	Metody weryfikacji zakładanych efektów kształcenia presentation, discussion	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Passing the seminar requires preparing and presenting a talk on the topic assigned by the instructor.	
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela	
	seminar	30 h
	Praca własna studenta	
	study of literature and additional materials	30 h
	presentation preparation	15 h
	Sumarycznie	
	Łączna liczba godzin	75 h
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminarium: Analiza problemów algorytmicznych Seminar: Analysis of algorithmic problems
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-AnPrbAlg
6.	Rodzaj przedmiotu/modułu Seminarium
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy oraz letni
11.	Forma zajęć i liczba godzin seminarium — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zalecane zrealizowane przedmioty • Logika dla informatyków • Wstęp do informatyki (Lub doświadczenie zdobyte podczas przygotowań/udziału w Olimpiadzie Informatycznej i podobnych konkursach.)
13.	Cele kształcenia dla przedmiotu Podstawowym celem przedmiotu jest omawianie zadań i technik pojawiających się na różnego rodzaju zawodach programistycznych (indywidualnych: Olimpiada Informatyczna, Potyczki Algorytmiczne, rundy Codeforces, drużynowych: AMPPZ, CERC, ICPC World Finals). Oprócz zapoznania się uczestników ze wzorcowymi rozwiązaniami celem jest także refleksja nad poprawnością/złożonością tych rozwiązań, na co (zwykle) nie ma czasu podczas samych zawodów.
14.	Treści programowe Podczas zajęć omawianie są (między innymi) zadania z zawodów: 1. Olimpiada Informatyczna. 2. Potyczki Algorytmiczne. 3. Rundy Codeforces. 4. Akademickie Mistrzostwa Polski w Programowaniu Zespołowym. 5. Central European Regional Contest. 6. ICPC World Finals. Oprócz samych zadań i ich rozwiązań uczestnicy przedstawiają ogólne techniki oraz triki przydatne w ich sprawnej implementacji.

15.	Zakładane efekty uczenia się	
	Wiedza	
	zna podstawowe metody projektowania i analizy algorytmów	K_W08
	Umiejętności	
	potrafi formułować precyzyjną specyfikację problemu oraz przedstawić formalny dowód poprawności i złożoności wybranego rozwiązania	K_U10
	potrafi przedstawić wybrane rozwiązanie i brać udział w dyskusji na jego temat	K_U11, K_U10
16.	Literatura obowiązkowa i zalecana Brak.	
17.	Metody weryfikacji zakładanych efektów uczenia się Prezentacja rozwiązania zadania, prezentacja ogólnej techniki.	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia seminarium konieczny jest aktywny udział w zajęciach (sugerowanie/omawianie zadań) lub przygotowanie i przedstawienie krótkiego wystąpienia na wybrany przez siebie temat.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	studiowanie omawianych zadań i literatury	45 godz.
	Sumarycznie	
	Łączna liczba godzin	75 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminarium: Blockchain i jego zastosowania Seminar on Blockchain and its applications								
2.	Dyscyplina informatyka								
3.	Język wykładowy polski								
4.	Jednostka prowadząca przedmiot Instytut Informatyki								
5.	Kod przedmiotu/modułu 28-INF-S-DSBLKCHN								
6.	Rodzaj przedmiotu/modułu Seminarium								
7.	Kierunek studiów (specjalność/specjalizacja) Data Science								
8.	Poziom studiów studia II 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 Ogólne obycie w świecie bezpieczeństwa IT.								
13.	Cele kształcenia dla przedmiotu Celem seminarium jest zgłębienie technologii Blockchain i obszarów jej zastosowania w znacznie szerszym kontekście niż tylko w kryptowalutach.								
14.	Treści programowe Tematyka seminarium jest związana z technologią Blockchain i obszarami jej zastosowania rozumianymi w znacznie szerszym kontekście niż tylko w kryptowalutach.								
15.	Zakładane efekty uczenia się Wiedza <table border="1"><tr><td>Zna dobrze wybrane zagadnienia technologii Blockchain</td><td>K_W08</td></tr></table> Umiejętności <table border="1"><tr><td>Umie przedstawić wybrane zagadnienie w przystępny i zrozumiały sposób</td><td>K_U11</td></tr><tr><td>Umie znaleźć i opracować odpowiednie materiały dla wybranego tematu</td><td>K_U12</td></tr><tr><td>Umie dobrać odpowiednie przykłady dla prezentowanego materiału</td><td>K_U11</td></tr></table>	Zna dobrze wybrane zagadnienia technologii Blockchain	K_W08	Umie przedstawić wybrane zagadnienie w przystępny i zrozumiały sposób	K_U11	Umie znaleźć i opracować odpowiednie materiały dla wybranego tematu	K_U12	Umie dobrać odpowiednie przykłady dla prezentowanego materiału	K_U11
Zna dobrze wybrane zagadnienia technologii Blockchain	K_W08								
Umie przedstawić wybrane zagadnienie w przystępny i zrozumiały sposób	K_U11								
Umie znaleźć i opracować odpowiednie materiały dla wybranego tematu	K_U12								
Umie dobrać odpowiednie przykłady dla prezentowanego materiału	K_U11								
16.	Literatura obowiązkowa i zalecana Artykuły dotyczące prezentowanych tematów.								
17.	Metody weryfikacji zakładanych efektów uczenia się Opracowanie i przedstawienie prezentacji na wybrany temat, dyskusja, rozmowa z prowadzącym								
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Zaliczenie na podstawie prezentacji z wybranego zagadnienia.								

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	przygotowanie raportu/prezentacji	15 godz.
	studiowanie tematyki wykładów i literatury	30 godz.
	Sumarycznie	
	Łączna liczba godzin	75 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminarium: Hakowanie dla każdego										
2.	Dyscyplina informatyka										
3.	Język wykładowy polski										
4.	Jednostka prowadząca przedmiot Instytut Informatyki										
5.	Kod przedmiotu/modułu 28-INF-S-DSGHack4All										
6.	Rodzaj przedmiotu/modułu Seminarium										
7.	Kierunek studiów (specjalność/specjalizacja) Data Science										
8.	Poziom studiów studia II 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										
13.	Cele przedmiotu Celem seminarium jest zapoznanie się ze światem etycznego hakowania. Przyjrzymy się popularnym podatnościom i ich wykorzystaniu do przełamywania zabezpieczeń. Zapoznamy się też z różnymi aktywnościami, gdzie etyczne hakowanie można wykorzystać, jak np. zawodu CTF czy programy bug bounty.										
14.	Treści programowe 1. Przegląd popularnych podatności i ich wykorzystanie 2. Zapoznanie się z odpowiednimi narzędziami 3. Zawody CTF i platformy, gdzie można próbować swoich sił 4. Programy bug bounty										
15.	Zakładane efekty uczenia się Wiedza <table border="1" data-bbox="225 1592 1430 1626"> <tr> <td>Zna dobrze wybrane zagadnienia z bezpieczeństwa systemów informatycznych</td><td>K_W08</td></tr> </table> Umiejętności <table border="1" data-bbox="225 1711 1430 1839"> <tr> <td>Umie zaprojektować rozwiązanie dla wybranego aspektu</td><td>K_U08</td></tr> <tr> <td>Umie przedstawić wybrane zagadnienie w przystępny i zrozumiały sposób</td><td>K_U08</td></tr> <tr> <td>Umie znaleźć i opracować odpowiednie materiały dla wybranego tematu</td><td>K_U08</td></tr> <tr> <td>Umie dobrać odpowiednie przykłady dla prezentowanego materiału</td><td>K_U08</td></tr> </table>	Zna dobrze wybrane zagadnienia z bezpieczeństwa systemów informatycznych	K_W08	Umie zaprojektować rozwiązanie dla wybranego aspektu	K_U08	Umie przedstawić wybrane zagadnienie w przystępny i zrozumiały sposób	K_U08	Umie znaleźć i opracować odpowiednie materiały dla wybranego tematu	K_U08	Umie dobrać odpowiednie przykłady dla prezentowanego materiału	K_U08
Zna dobrze wybrane zagadnienia z bezpieczeństwa systemów informatycznych	K_W08										
Umie zaprojektować rozwiązanie dla wybranego aspektu	K_U08										
Umie przedstawić wybrane zagadnienie w przystępny i zrozumiały sposób	K_U08										
Umie znaleźć i opracować odpowiednie materiały dla wybranego tematu	K_U08										
Umie dobrać odpowiednie przykłady dla prezentowanego materiału	K_U08										
16.	Literatura obowiązkowa i zalecana Przegląd artykułów w Sieci z wybranych zagadnień										
17.	Metody weryfikacji zakładanych efektów kształcenia prezentacja, dyskusja										
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Zaliczenie seminarium wymaga przygotowania i wygłoszenia referatu na przydzielony przez prowadzącego temat.										

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	przygotowanie raportu/prezentacji	15 godz.
	studiowanie tematyki wykładów i literatury	30 godz.
	Sumarycznie	
	Łączna liczba godzin	75 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminarium: Kompetetywna Sztuczna Inteligencja Seminar: Competitive Artificial Intelligence
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DSCompeAI
6.	Rodzaj przedmiotu/modułu Seminarium
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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 Znajomość tematyki omawianej na przedmiocie Sztuczna Inteligencja (wymagane) oraz Artificial Intelligence for Games (zalecane).
13.	Cele kształcenia dla przedmiotu Znajomość istniejących zawodów programistycznych AI oraz stosowanych w nich algorytmów. Umiejętność wyboru odpowiedniego podejścia w zależności od struktury zadanego problemu.
14.	Treści programowe Zaawansowane algorytmy sztucznej inteligencji oraz ich rozmaite warianty, rozszerzenia i możliwości optymalizacji (MiniMax, MCTS, RHEA, BeamSearch/ChokudaiSearch, Transposition Tables, Opening Books, Pathfinding, ...). Zawody kompetetywnej sztucznej inteligencji, platformy oraz protokoły (CodinGame, IEEE COG competitions, AI Cup, Battlecode, Bomberlnd, Terminal, Lux AI, ...).

15.	Zakładane efekty uczenia się	
	Wiedza	
	Zna najnowsze badania w dziedzinie sztucznej inteligencji w grach, orientuje się w zaawansowanych algorytmach AI stosowanych w grach komputerowych oraz zna najważniejsze odbywające się w tematyce zawody	K_W08
	Posiada przeglądową wiedzę na temat aktualnie prowadzonych badań w dziedzinie Game AI	K_W08
	Umiejętności	
	Potrafi samodzielnie zrozumieć pracę naukową i zgłębić wybrany temat wyszukując informacje	K_U12
16.	Potrafi przygotować i wygłosić prezentację dotyczącą wybranego tematu	K_U11
	Kompetencje społeczne	
	Potrafi przeprowadzić krytyczną analizę wybranego tematu związanego z tematyką gier	K_K01
	Potrafi brać udział w dyskusji, formułując swoje opinie dotyczące prezentowanego tematu	K_K03
17.	Literatura obowiązkowa i zalecana	
18.	Wybrane prace naukowe z dziedziny, blogi uczestników zawodów, opisy zwycięskich programów, dokumentacja techniczna konkursów.	
19.	Metody weryfikacji zakładanych efektów uczenia się	
20.	- Opracowanie i przedstawienie prezentacji na zadany temat - Dyskusja	
21.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
22.	- Opracowanie wybranego tematu. - Ocena wystąpienia seminaryjnego na temat zadanego materiału. - Obecność na zajęciach i udział w dyskusjach.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	studiowanie wybranego temtu	25 godz.
	przygotowanie prezentacji	15 godz.
	Sumarycznie	
	Łączna liczba godzin	70 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminarium: Wielkie modele językowe Seminar: Large Language Models
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DSLLMs
6.	Rodzaj przedmiotu/modułu Seminarium
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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 Niezbędne kompetencje: • Znajomość koncepcji uczenia maszynowego • Ogólna znajomość sieci neuronowych, w szczególności podstawowa wiedza na temat sieci typu transformer
13.	Cele przedmiotu Podstawowym celem kursu jest zapoznanie studentów z możliwościami i ograniczeniami sieci neuronowych modelujących język i próba odpowiedzi na pytanie, do jakiego stopnia technologia modeli językowych może być uznawana za załączek ogólnej sztucznej inteligencji
14.	Treści programowe 1. Sieci transformer i autoregresywne modele językowe 2. Różne sposoby promptingu i fine tuningu 3. Modelowanie sekwencji innych niż pochodzące z języka 4. Modele wielomodalne 5. Tworzenie agentów korzystających z modeli językowych 6. Łańcuchy myśli i zagadnienia pokrewne (chain-of-thoughts, tree-of-thoughts, ...)

15.	Zakładane efekty uczenia się Wiedza	
	rozumie podstawowe podejścia modeli językowych	K_W05, K_W07, K_W08
	rozumie problemy związane z LLM, w szczególności rozumie, jak modelowanie języka może służyć do rozwiązywania innych problemów sztucznej inteligencji (wnioskowanie, planowanie, przeszukiwanie)	K_W05, K_W07, K_W08
	posiada przeglądową wiedzę dotyczącą różnych wariantów LLM	K_W05, K_W07, K_W08
	Umiejętności	
	umie formułować zagadnienia praktyczne jako zadania modelowania specyficznego języka	K_U02
	umie stosować i modyfikować podstawowe idee LLM	K_U08
	umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy	K_U11
	Kompetencje społeczne	
	rozumie znaczenie modelowania języka dla funkcjonowania współczesnego społeczeństwa	K_K02
16.	Literatura obowiązkowa i zalecana • Publikacje z konferencji ACL, NeurIPS, ICRL, i innych • Publikacje wymieniane w https://www.marktechpost.com/ w kategorii LLM (oraz w innych agregatorach newsów naukowych)	
17.	Metody weryfikacji zakładanych efektów kształcenia • Prezentacje prac naukowych • Prezentacje swoich eksperymentów	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia przedmiotu wymagane jest przygotowanie prezentacji na temat wybrany przez studenta (z puli tematów zaproponowanych przez prowadzącego)	
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	studiowanie literatury i materiałów dodatkowych	30 godz.
	przygotowanie prezentacji	15 godz.
	Sumarycznie	
	Łączna liczba godzin	75 godz.
	Liczba punktów ECTS	3

5 Projekty zespołowe

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 zespołowy
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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 • Przedmiot przedstawiający tematykę sieci neuronowych. Niezbędne kompetencje • 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: • 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: • odpowiednią modularność kodu • reprodukowalność eksperymentów poprzez zachowanie wersji kodu, zbiorów danych (dataset registry) i modeli (model registry). Realizacja zadań programistycznych będzie wykorzystywała następujące narzędzia (ang. framework): • 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.	Zakładane efekty uczenia się Wiedza <table border="1"> <tr> <td>ma pogłębioną wiedzę z zakresu głębokich sieci neuronowych. Dobrze zna rolę i znaczenie ich architektury w modelowaniu danych</td><td>K_W01, K_W08</td></tr> <tr> <td>zna zaawansowane techniki obliczeniowe z wykorzystaniem procesorów CPU i GPU, a także rozumie ich ograniczenia</td><td>K_W03, K_W08</td></tr> <tr> <td>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)</td><td>K_W08</td></tr> </table> Umiejętności <table border="1"> <tr> <td>potrafi zastosować wiedzę o sieciach neuronowych, do analizowania i rozwiązywania problemów klasyfikacji danych wizyjnych i tekstowych</td><td>K_U02, K_U03</td></tr> <tr> <td>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</td><td>K_U04, K_U06</td></tr> <tr> <td>posługuje się cyfrową reprezentacją danych: liczb, obrazów i tekstów</td><td>K_U03</td></tr> <tr> <td>potrafi implementować programy w zespole</td><td>K_U13</td></tr> </table> Kompetencje społeczne <table border="1"> <tr> <td>jest świadom możliwości popełniania błędów przez siebie i innych. Wykazuje rozważny krytycyzm wobec odbieranych treści oraz otrzymywanych wyników, zwłaszcza przy interpretacji wyników uzyskiwanych z użyciem modeli głębokich sieci neuronowych</td><td>K_K01</td></tr> </table>	ma pogłębioną wiedzę z zakresu głębokich sieci neuronowych. Dobrze zna rolę i znaczenie ich architektury w modelowaniu danych	K_W01, K_W08	zna zaawansowane techniki obliczeniowe z wykorzystaniem procesorów CPU i GPU, a także rozumie ich ograniczenia	K_W03, K_W08	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)	K_W08	potrafi zastosować wiedzę o sieciach neuronowych, do analizowania i rozwiązywania problemów klasyfikacji danych wizyjnych i tekstowych	K_U02, K_U03	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	K_U04, K_U06	posługuje się cyfrową reprezentacją danych: liczb, obrazów i tekstów	K_U03	potrafi implementować programy w zespole	K_U13	jest świadom możliwości popełniania błędów przez siebie i innych. Wykazuje rozważny krytycyzm wobec odbieranych treści oraz otrzymywanych wyników, zwłaszcza przy interpretacji wyników uzyskiwanych z użyciem modeli głębokich sieci neuronowych	K_K01
ma pogłębioną wiedzę z zakresu głębokich sieci neuronowych. Dobrze zna rolę i znaczenie ich architektury w modelowaniu danych	K_W01, K_W08																
zna zaawansowane techniki obliczeniowe z wykorzystaniem procesorów CPU i GPU, a także rozumie ich ograniczenia	K_W03, K_W08																
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)	K_W08																
potrafi zastosować wiedzę o sieciach neuronowych, do analizowania i rozwiązywania problemów klasyfikacji danych wizyjnych i tekstowych	K_U02, K_U03																
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	K_U04, K_U06																
posługuje się cyfrową reprezentacją danych: liczb, obrazów i tekstów	K_U03																
potrafi implementować programy w zespole	K_U13																
jest świadom możliwości popełniania błędów przez siebie i innych. Wykazuje rozważny krytycyzm wobec odbieranych treści oraz otrzymywanych wyników, zwłaszcza przy interpretacji wyników uzyskiwanych z użyciem modeli głębokich sieci neuronowych	K_K01																
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.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	Praca własna studenta	
	studiowanie literatury	40 godz.
	praca nad projektem	80 godz.
	przygotowanie raportu/prezentacji	10 godz.
	Sumarycznie	
	Łączna liczba godzin	160 godz.
	Liczba punktów ECTS	6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt: Machine Learning for Recommender Systems Project: Machine Learning for Recommender Systems
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu ??
6.	Rodzaj przedmiotu/modułu Projekt zespołowy
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin pracowni – 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 przedmiotu Celem przedmiotu jest nabycie/doskonalenie umiejętności pracy zespołowej oraz rozszerzenie wiedzy z algorytmów uczenia maszynowego dla systemów rekomendujących.
14.	Treści programowe 1. Podstawowe algorytmy klasycznych systemów rekomendujących, takie jak Collaborative Filtering i Matrix Factorization. 2. Ich rozszerzenia, takie jak Factorization Machines, Neural Collaborative Filtering, Neural Graph Collaborative Filtering. 3. Podstawowe algorytmy Session-based Recommender Systems, takie jak Session-based Graph Neural Networks, Target Attentive Graph Neural Networks. 4. Ich rozszerzenia oparte na współczesnych algorytmach uczenia maszynowego, głębokich sieciach neuronowych i grafowych sieciach neuronowych.

15.	Zakładane efekty uczenia się	
	Wiedza	
	rozumie podstawowe podejścia systemów rekomendujących	K_W02, K_W08
	rozumie problemy związane z systemami rekomendującymi	K_W02, K_W08
	posiada przeglądową wiedzę o różnych podejściach do systemów rekomendujących, w tym dla rekomendacji opartych na macierzy ocen i dla rekomendacji opartych na zapisie sesji	K_W08
	zna wybrane podstawowe algorytmy systemów rekomendujących	K_W06, K_W08
	Umiejętności	
	umie konstruować modele systemów rekomendujących odpowiadające wymaganiom praktycznym związanym z potrzebami świata rzeczywistego, w tym rekomendacjami e-commerce, rekomendacjami treści, itp.	K_U06, K_U07
	umie stosować i modyfikować podstawowe algorytmy uczenia systemów rekomendujących	K_U08
	umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy	K_U09, K_U10, K_U11
	Kompetencje społeczne	
	rozumie znaczenie systemów rekomendujących dla funkcjonowania współczesnego społeczeństwa, w tym korzyści i zagrożenia wynikające z ich stosowania	K_K02
16.	Literatura obowiązkowa i zalecana • Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor, Recommender Systems Handbook, Springer, 2011. • artykuły naukowe dot. systemów rekomendujących, m.in. z konferencji NIPS, WWW, SIGIR, RecSys, itp.	
17.	Metody weryfikacji zakładanych efektów kształcenia prezentacja, dyskusja, (grupowe, wspólne) projektowanie i programowanie systemu informatycznego	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Zaliczenie projektu wymaga regularnej i rzetelnej pracy w projekcie, kontrolowanej na bieżąco przez prowadzącego, realizacji (indywidualnie lub grupowo) zadań projektowych przydzielanych na bieżąco przez prowadzącego, regularnego raportowania postępu w pracach, prezentacji (częściowych lub całościowych) wyników przydzielonych zadań, regularnego udziału w dyskusjach.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	Praca własna studenta	
	studiowanie literatury i materiałów dodatkowych	15 godz.
	przygotowanie dokumentacji, prezentacji, raportów	10 godz.
	wykonywanie zadań projektowych	45 godz.
	Sumarycznie	
	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 zespołowy
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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: • Sztuczna inteligencja Niezbędne kompetencje: • 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 1. Klasyczne algorytmy przeszukiwań, alfa-beta obcięcia i MCTS 2. Sieć neuronowa, która uczy się poprawnej, choć rzecz jasna przybliżonej, ewaluacji stanów 3. Implementacja klasycznej metody przeszukiwania wzbogaconą o deep learningową funkcję ewaluacji stanów 4. 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_W04, K_W08 K_W08 K_W05, K_W07
	Umiejętności	
16.	potrafi zaimplementować silnik gry w szachy potrafi stosować sieci neuronowe do reprezentacji strategii gry w szachy umie tworzyć oprogramowanie współpracując w grupie	K_U02, K_U06 K_U02, K_U08 K_U13
	Literatura obowiązkowa i zalecana	
17.	- Chess Algorithms, Noah Caplinger - Neural Networks for Chess, Dominik Klein	
	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 modulem silnika	65 godz.
	Sumarycznie	
	Łączna liczba godzin	95 godz.
	Liczba punktów ECTS	4

6 Humanistyczno-społeczne

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) IData Science
8.	Poziom studiów studia II 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
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 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: pomieszczenie porządków logicznych i przyczynowo – 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 <table border="1"> <tr> <td>zna podstawowe idee określające nurt post-humanizmu</td><td>K_W09</td></tr> <tr> <td>zna podstawy rozumienia kultury, relacji społecznych umysłu i świadomości w kontekście różnych nurtów nauk kognitywnych</td><td>K_W09</td></tr> </table>	zna podstawowe idee określające nurt post-humanizmu	K_W09	zna podstawy rozumienia kultury, relacji społecznych umysłu i świadomości w kontekście różnych nurtów nauk kognitywnych	K_W09
zna podstawowe idee określające nurt post-humanizmu	K_W09				
zna podstawy rozumienia kultury, relacji społecznych umysłu i świadomości w kontekście różnych nurtów nauk kognitywnych	K_W09				
16.	Literatura obowiązkowa i zalecana Literatura obowiązkowa: 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 • 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	
	wykład	30 godz.
	ćwiczenia	30 godz.
	Praca własna studenta	
	studiowanie tematyki wykładów i literatury	40 godz.
	przygotowanie do egzaminu	20 godz.
	Sumarycznie	
	Łączna liczba godzin	120 godz.
	Liczba punktów ECTS	5

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim O ekonomii i gospodarce inaczej Another View of Economy
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DHSEGI
6.	Rodzaj przedmiotu/modułu Humanistyczno-społeczny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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
13.	Cele przedmiotu Opis zjawisk gospodarczych w kontekście procesów globalizacyjnych i zagrożenia kryzysem ekonomicznym w skali świata. Przedstawienie innego, nieortodoksyjnego spojrzenia na ekonomię i zjawiska gospodarcze.
14.	Treści programowe 1. Ekonomia: nauka ścisła, społeczna, czy historyczna? 2. Dyskryminacja kobiet na rynku pracy: czy faktycznie występuje? 3. Sztuczna inteligencja a przyszłość gospodarki: co uda się zastąpić, a czego nie? 4. Hiperinflacja i katastrofy monetarne: dlaczego nieraz taniej było palić banknotami w piecu. 5. Inflacja i deflacja: dlaczego mimo szczelnego zaryglowania portfela ktoś ciągle wyciąga mi z niego pieniądze? 6. Wielkie Kryzys: czyli jak ekonomia wyjaśni dyktatury i drugą wojnę światową? 7. Pandemia i ekonomiczne skutki: niepozorne konsekwencje siedzenia w domu 8. Prohibicja narkotykowa i jej efekty: czy tę wojnę da się wygrać? 9. Reklama - marnotrawstwo czy nieunikniony aspekt konkurencji? 10. Ekonomia Thanosa (Malthusa) - czy śmierć połowy ludzi może cokolwiek poprawić? 11. Niewolnictwo i agresywny imperializm: czy to naprawdę taki świetny interes wsadzić innych do klatek? 12. Rewolucja przemysłowa, dlaczego jesteśmy tak bogaci? 13. Sektor publiczny - czyli co robi państwo i skąd się biorą uzasadnienia tego, co ma robić państwo? 14. Wolny handel - czy zboże z Ukrainy nas zniszczy? 15. Patenty, prawa autorskie i własność intelektualna - czy można posiadać idee? 16. Kryzys w strefie euro. Czy niemieckie oszczędności zniszczyły Grecję? 17. Jak mogłaby funkcjonować gospodarka oparta na kryptowalutach? 18. Spory o nieruchomości. Czy ekolodzy powinni popierać gęsta zabudowę i dlaczego mieszkania są drogie? 19. Migracje, czyli jak podwoić dochód światowy w kilka lat.

15.	Zakładane efekty uczenia się Wiedza zna podstawowe pojęcia z zakresu ekonomii zna historyczne oraz demograficzne uwarunkowania rozwoju gospodarczego		K_W09 K_W09
16.	Literatura obowiązkowa i zalecana - Robert P. Murphy: Niepoprawny politycznie przewodnik po kapitalizmie - John Tamny: Ekonomia zdrowego rozsądku		
17.	Metody weryfikacji zakładanych efektów kształcenia egzamin pisemny, dyskusja podczas ćwiczeń		
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Zaliczenie przedmiotu jest na podstawie egzaminu pisemnego. Ocena z ćwiczeń będzie głównie odpowiadała jakości przygotowania się każdego studenta do ćwiczeń i jego aktywności na ćwiczeniach. Do każdego spotkania podana będzie literatura (jeden, dwa artykuły, fragmenty książek, itp.), którą studenci będą mieli do przeczytania przed każdymi ćwiczeniami. To będzie podstawą do dyskusji na ćwiczeniach.		
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela ćwiczenia wykład		30 godz. 30 godz.
	Praca własna studenta przygotowanie do egzaminu przygotowanie do ćwiczeń samodzielne studiowanie literatury		10 godz. 25 godz. 30 godz.
	Sumarycznie Łączna liczba godzin Liczba punktów ECTS		125 godz. 5

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Wstęp do filozofii umysłu i sztucznej inteligencji Introduction to the philosophy of mind and artificial intelligence
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DHSFilozAI
6.	Rodzaj przedmiotu/modułu Humanistyczno-społeczny
7.	Kierunek studiów (specjalność/specjalizacja) Data Science
8.	Poziom studiów studia II 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
13.	Cele przedmiotu Wykład stanowi wprowadzenie do elementarnych zagadnień współczesnej filozofii umysłu oraz filozofii sztucznej inteligencji. W jego trakcie studenci będą mogli zapoznać się z najważniejszymi problemami formułowanymi w obu tych obszarach i próbami ich rozwiązania w obrębie najistotniejszych zagadnień, które zaprezentowane zostaną przez pryzmat klasycznych tekstów dotyczących filozofii umysłu i sztucznej inteligencji.
14.	Treści programowe 1. Kwestia natury umysłu oraz różnorakich dotyczących tego teorii (np. dualizm, behawioryzm, teoria identyczności, funkcjonalizm). 2. Zagadnienie problemu umysł-ciało, natura świadomości i kwestia świadomości sztucznej inteligencji. 3. Spór o istnienie qualiów, znaczenie pojęć emergencji i superweniencji w filozofii umysłu. 4. Etyczne aspekty badań nad sztuczną inteligencją. 5. Problem wolnej woli. 6. Problem reprezentacyjnego modelu umysłu. 7. Koncepcje rozszerzonego umysłu i związku sztucznej inteligencji z obliczeniowymi teoriami umysłu.

15.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>potrafi wskazać najważniejsze problemy filozofii umysłu i sztucznej inteligencji przyswoił sobie odpowiednią dla filozofii umysłu terminologię i potrafi dokonywać dystynkcji w obszarze filozoficznej problematyki badań nad umysłem, sztuczną inteligencją i świadomością</td><td>K_W09 K_W09</td></tr> <tr> <td>jest zaznajomiony jest z głównymi argumentami stron w sporach w obrębie filozofii umysłu i potrafi samodzielnie je sformułować</td><td>K_W09</td></tr> </table> Umiejętności <table> <tr> <td>potrafi odróżnić specyfikę filozoficznych badań nad umysłem i sztuczną inteligencją od badań w dziedzinach pokrewnych, jak np. psychologia, neuronauki, informatyka, kognitywistyka, oraz dostrzega łączące je związki w twórczy sposób potrafi poddać analizie wybrany przez siebie problem w obszarze filozofii umysłu i sztucznej inteligencji</td><td></td></tr> </table>	potrafi wskazać najważniejsze problemy filozofii umysłu i sztucznej inteligencji przyswoił sobie odpowiednią dla filozofii umysłu terminologię i potrafi dokonywać dystynkcji w obszarze filozoficznej problematyki badań nad umysłem, sztuczną inteligencją i świadomością	K_W09 K_W09	jest zaznajomiony jest z głównymi argumentami stron w sporach w obrębie filozofii umysłu i potrafi samodzielnie je sformułować	K_W09	potrafi odróżnić specyfikę filozoficznych badań nad umysłem i sztuczną inteligencją od badań w dziedzinach pokrewnych, jak np. psychologia, neuronauki, informatyka, kognitywistyka, oraz dostrzega łączące je związki w twórczy sposób potrafi poddać analizie wybrany przez siebie problem w obszarze filozofii umysłu i sztucznej inteligencji					
potrafi wskazać najważniejsze problemy filozofii umysłu i sztucznej inteligencji przyswoił sobie odpowiednią dla filozofii umysłu terminologię i potrafi dokonywać dystynkcji w obszarze filozoficznej problematyki badań nad umysłem, sztuczną inteligencją i świadomością	K_W09 K_W09										
jest zaznajomiony jest z głównymi argumentami stron w sporach w obrębie filozofii umysłu i potrafi samodzielnie je sformułować	K_W09										
potrafi odróżnić specyfikę filozoficznych badań nad umysłem i sztuczną inteligencją od badań w dziedzinach pokrewnych, jak np. psychologia, neuronauki, informatyka, kognitywistyka, oraz dostrzega łączące je związki w twórczy sposób potrafi poddać analizie wybrany przez siebie problem w obszarze filozofii umysłu i sztucznej inteligencji											
16.	Literatura obowiązkowa i zalecana 1. G. Ryle (1970), Co to jest umysł?, przeł. W. Marciszewski, PWN 2. D. Armstrong (1982), Materialistyczna teoria umysłu, przeł. H. Krahelska, PWN 3. Filozofia umysłu (1995), B. Chwedeńczuk (red.), Aletheia 4. D. Chalmers (2010), Świadomy umysł, przeł. M. Miłkowski, PWN 5. Przewodnik po filozofii umysłu (2012), M. Miłkowski, R. Poczubut (red.), WAM 6. T. Skalski (2013), Trzy problemy współczesnej filozofii umysłu, Wydawnictwo Uniwersytetu Łódzkiego 7. D. Dennett (2016), Świadomość, przeł. E. Stokłosa, Copernicus Center Press 8. Analityczna metafizyka umysłu (2008), M. Miłkowski, R. Poczubut (red.), Wydawnictwo IFiS PAN 9. N. Bostrom (2016), Superinteligencja. Scenariusze, strategie, zagrożenia, przeł. D. Konowrocka-Sawa, Helion 10. M. Boden (2020), Sztuczna inteligencja. Jej natura i przyszłość, przeł. T. Sieczkowski, Wydawnictwo Uniwersytetu Łódzkiego 11. S. Schneider (2021), Świadome maszyny. Sztuczna inteligencja i projektowanie umysłów, przeł. J. Bednarek, PWN										
17.	Metody weryfikacji zakładanych efektów kształcenia egzamin pisemny										
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Egzamin ma formę pisemną.										
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>wykład</td><td>30 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>samodzielne studiowanie literatury</td><td>30 godz.</td></tr> <tr> <td>przygotowanie do egzaminu</td><td>15 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>75 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>3</td></tr> </table>	wykład	30 godz.	samodzielne studiowanie literatury	30 godz.	przygotowanie do egzaminu	15 godz.	Łączna liczba godzin	75 godz.	Liczba punktów ECTS	3
wykład	30 godz.										
samodzielne studiowanie literatury	30 godz.										
przygotowanie do egzaminu	15 godz.										
Łączna liczba godzin	75 godz.										
Liczba punktów ECTS	3										