

# Course Programming in C++

## Exercise List 4

Deadline: 07.04.2016

Topic of this exercise are the use of defined operators, and the use of private fields.

1. In class **rational** of Task 2, make the fields **num** and **denum** private. Add getters for **num** and **denum**, and rewrite the rest of the program so that it compiles again.
2. Extend the **stack** class of Task 3 with operators

```
double operator [] ( size_t i ) const;
double& operator [] ( size_t i );
```

The top of stack has index 0 and the element on the bottom of the stack has index `size() - 1`.

If you want the operators to check bounds, throw a `std::runtime_error`. Don't return a nonsense value.

Make tests in which both operators are used.

3. Add operators

```
void operator += ( double ); // Same as push( );
void operator += ( const stack& s );
// Be careful with self assignment!
// The second operator pushes all elements of s.
```

4. Extend the **stack** class with `stack operator + ( const stack& s1, const stack& s2 )`, `stack operator + ( stack s1, const stack& s2 )` or some other variation of `operator +`.

You can make use of `operator +=` or use `operator []`.

(The addition operators on stacks are unnatural, but we need to practice the use of these operators in some way.)

5. Check for the absence of memory leaks, using code that contains the addition operator, the new assignment operators. Use **valgrind**. Make sure that a self assignment `s+=s`; is included in the tests.