

Exercise sheet 11

To be handed in on 12.07.2024.

Exercise 1 [C++ classes]

(10 bonus pts.)

- (i) Implement a class `Complex`, representing a complex number $z \in \mathbb{C}$. This class should have a constructor `Complex(double re, double im)`, which initializes the `private` attributes `double real` and `double imag` corresponding to $\operatorname{Re}(z)$ and $\operatorname{Im}(z)$, respectively. Add so-called *getters* `getReal()` and `getImag()` to the class `Complex`, i.e. member functions, which return `real` and `imag`.

Hint: Do not forget the `const` keyword for `getReal()` and `getImag()` at the end of their declarations.

- (ii) Test the following lines of code in your `main` function:

```
Complex c1(1.0, 1.0);
Complex c2(c1);
printf("%.2f %.2f\n", c2.getReal(), c2.getImag());
```

You did not define a copy-constructor `Complex(const Complex &c)` and yet, you were able to make a copy of `c1`. Explain, why this is the case.

- (iii) Add a member function to your class, which returns $|z|^2 = \operatorname{Re}(z)^2 + \operatorname{Im}(z)^2$ and test it in your `main` function for $z = \cos(\pi/4) + i\sin(\pi/4)$.

- (iv) Implement the function `exp` for the class `Complex`, i.e. define a function `Complex exp(const Complex &c)` outside the class `Complex` such, that a `Complex` object corresponding to $e^z = e^{\operatorname{Re}(z)}(\cos(\operatorname{Im}(z)) + i\sin(\operatorname{Im}(z)))$, $z \in \mathbb{C}$ is returned. Explain the concept of “overloading” in the context of this example. Test this function by adding the following lines of code to the main function:

```
Complex c3 = exp(c2);
printf("%.2f %.2f\n", c3.getReal(), c3.getImag());
```

Discuss, why the `const` keyword at the end of the declarations of `getReal` and `getImag` is essential. Explore, what happens, when you omit the `const` keyword.

- (v) Overload the `+` operator via `Complex operator+(const Complex &c1, const Complex &c2)` and consider the following additional lines of code in your `main` function:

```
c1 = c1 + 1.0;
printf("%.2f %.2f\n", c2.getReal(), c2.getImag());
```

Explain, why this code does not compile.

Now, define an additional constructor `Complex(double x)`, which initializes the object with `real = x` and `imag = 0.0` and run the code above again. The code should compile and run now. Explain why.