

Object-oriented Programming

Exam Sheet No. 3

Date: December 13— Due: December 20

Submission by Email

It is *also* allowed to submit the solutions for this exam sheet by email. If you want to submit by email follow these instructions:

- Send the email to: `carsten.gutwenger@cs.tu-dortmund.de`
- The subject of you email must be: OOP Exam 3
- Your email text must contain your name and matriculation number.
- Submit only the source files (.cpp-files) and not the whole project.
- Attach each source file to your email; the name of the file must reflect the exam it solves, e.g., `exam_3_1.cpp` should contain your solutions to Exam 3.1.
- The deadline for submission is December 20, 10:30 am.

Exam 3.1

Implement a class `Matrix`, which shall represent an $n \times m$ matrix of doubles.

- Write a constructor that takes n (the number of rows) and m (the number of columns) as parameters and initializes all elements to 0.
- Implement a member function for accessing the elements in the matrix, e.g.
`double &at(int i, int j);`
- Overload the `+`-operator (for adding to matrices) and the output operator for your `Matrix` class.

Write a small test program that demonstrates your implementations.

Exam 3.2

A *complex number* is a number consisting of a real part and an imaginary part, usually written as $a + bi$ where i is the imaginary unit with $i^2 = -1$. Read the Wikipedia article for more information: http://en.wikipedia.org/wiki/Complex_number

Implement a new data type `complex_number` which provides the following functionality:

- Data members `Re` and `Im`, both of type `double`, representing the real and imaginary part, respectively.
- A default constructor, a constructor with one parameter (the real part, the imaginary part shall be set to 0), and a constructor with two parameters (the real and the imaginary part).
- an output operator that prints a complex number as $a + bi$ if the imaginary part is non-zero, and as a usual floating point number otherwise.
- Overload the `+`-operator such that it is possible to add two complex numbers, as well as any combination of a complex number and a `double`.
- Overload the `*`-operator such that it is possible to multiply two complex numbers, as well as any combination of a complex number and a `double`.

Exam 3.3

You are given the following data structure for a record storing information about a student:

```
struct Student {
    std::string name;
    int         mat_no;
    std::map<std::string, int> grades;
};
```

The map `grades` stores for each student its courses and grades in these courses.

(a) Add a member function

```
bool add_grade(std::string course, int grade)
```

to `Student` which adds a new grade to the student's record. Take care that no grade is overwritten and that the grade entered is between 1 and 5. Return `true` if a new grade was added, `false` otherwise.

(b) Add a member function

```
void print_record(std::ostream &os)
```

to `Student` which prints the student's name, matriculation number, and all the courses the student has finished (their names and grades) to `os`.

Write a small test program that demonstrates the functionality provided by `Student`.