AGH University of Krakow | Faculty of Mechanical Engineering and Robotics
| Department of Robotics and Mechatronics

MECHATRONIC ENGINEERING | I cycle | semester VI | 2025/2026

Object oriented programming and software engineering

Instruction X:

Mini-project realization

You will learn:

How to design and implement a complete C++ application by combining multiple concepts introduced throughout the course.

You will learn how to structure a program using classes and multiple files, how to design interactions between objects, and how to manage and process data using STL containers and algorithms. You will also practice handling errors, working with files, and implementing more advanced features such as inheritance, polymorphism, and templates.

By completing this project, you will gain experience in building a functional system from scratch, similar to real engineering and programming tasks.

Course supervisor:
dr inż. Lucjan Miękina, miekina@agh.edu.pl

Instruction author:
mgr inż. Joanna Koszyk, jkoszyk@agh.edu.pl

1. Initial information

During these laboratory classes, you will work in pairs to develop a complete C++ application based on one of the chosen project topics.

The goal of the mini-project is to apply the concepts learned during the course, including object-oriented programming (classes, encapsulation, inheritance, polymorphism), memory management (references, pointers, dynamic memory), templates and generic programming, STL containers, algorithms, and iterators, file handling and error handling

The project should represent a simple but complete system instead of a collection of unrelated functions.

2. Grading requirements

The final grade depends on the features implemented in the mini-project and the correct use of C++ concepts.

GRADE 3.0

The project must contain:

- at least one class with attributes and methods
- working main functionality (program runs without crashing)
- use of functions (logic not placed only in main)
- processing a small set of data (e.g., a few elements)
- basic input/output (console interaction or simple display)

GRADE 3.5

The project must contain everything from 3.0 and additionally:

- at least two classes with different responsibilities
- encapsulation (private attributes, public methods)
- use of at least one STL container (std::vector, std::list, etc.)
- storing and processing multiple elements
- at least one data operation (e.g. searching, filtering, simple calculation)

GRADE 4.0

The project must contain everything from 3.5 and additionally:

- correct use of constructors and destructors
- interaction between objects (one class uses another)
- use of references or pointers in functions
- use of at least one STL algorithm, such as std::sort, std::find, std::count
- handling basic edge cases: empty data, invalid input

GRADE 4.5

The project must contain everything from 4.0 and additionally:

- separation into multiple files (.h and .cpp)

- use of inheritance or polymorphism: at least one base class and at least one derived class
- use of dynamic memory (new / delete) or clear use of pointers where appropriate
- use of iterators explicitly (not only range-based loops)
- use of additional STL containers (e.g., std::set, std::map)
- at least one component designed to be reusable (e.g., general-purpose class)

GRADE 5.0

The project must contain everything from 4.5 and additionally:

- use of polymorphism with base class pointers or references
- use of class and function templates
- use of multiple STL algorithms in meaningful context
- integration of containers, algorithms and iterators
- basic file handling (saving and/or loading data)
- use of exception handling (try / catch)
- improved program logic
- better system design