

C++ PROGRAMMING

1. **Course number and name:** 020PCPES2 C++ Programming
2. **Credits and contact hours:** 4 ECTS credits, 2x1:15 contact hours per week
3. **Name(s) of instructor(s) or course coordinator(s):** Ahmad Audi
4. **Instructional materials:** PowerPoint slides; lab experiments; Coursera links

Textbook/Reference:

- Robert Lafore, Object-Oriented Programming in C++, 4th edition, SAMS.

5. Specific course information

a. Catalog description:

Structure of a C++ program (declarations, statements, literals, operators), control statements (conditional statements and loops), functions, arrays, structures. Object-oriented programming: Classes and objects, construction, encapsulation, inheritance, virtual functions, abstract classes and polymorphism, operator overloading, exception handling, file handling, generic programming with templates, the Standard Template Library (STL), graphical interfaces with Qt.

b. Prerequisite: Programming 2 (020IF2NI3 or 020IF2CI3).

c. Required for ME students.

6. Educational objectives for the course

a. Specific outcomes of instruction:

- Implement a procedural C program.
- Implement an object-oriented C++ program in congruence with current best practices with regards to correctness, maintainability and performance.
- Analyze a C++ program with regards to its compliance to a given specification.
- Design and implement a C++ program to solve a complex problem.
- Evaluate the maintainability of C++ code and propose appropriate improvements.

b. PI addressed by the course:

PI	1.3	2.3	2.4	2.5	6.2	6.4	7.1	7.2
Covered	x	x	x	x	x	x	x	x
Assessed	x	x	x	x	x	x	x	x

7. Brief list of topics to be covered

- Introduction to C/C++ programming. Typed variable declarations, keyboard input, screen output, expressions (2 Lectures).
- Conditional branching, loops, and iterations (2 Lectures).
- Functions, parameter passing, overloading (3 Lectures).
- C-style arrays and C++ arrays (3 Lectures).
- Pointers and dynamic memory allocation (3 Lectures).
- Lab 1: functions, arrays, and pointers. Debugging (2 Lectures).
- Introduction to Object-Oriented Programming. Abstraction and Encapsulation. Classes and instances (2 Lectures).
- Visibility: private and public access, constructors and destructors, methods: actions and predicates, copying (2 Lectures).
- Lab 2: Classes (2 Lectures).
- Basic inheritance: concept, static link resolution and hiding, protected access, constructors and destructors, deep copying (3 Lectures)
- Basic polymorphism: dynamic link resolution and virtual functions, substitutions, abstract classes, and heterogeneous collections (2 Lectures).
- Lab 3: Inheritance and Polymorphism (2 Lectures).