



## Course Syllabus

<b>Course Code</b>	<b>Course Title</b>	<b>ECTS Credits</b>
COMP-111	Programming Principles I	6
<b>Prerequisites</b>	<b>Department</b>	<b>Semester</b>
None	Computer Science	Fall, Spring
<b>Type of Course</b>	<b>Field</b>	<b>Language of Instruction</b>
Compulsory	Computer Science	English/Greek
<b>Level of Course</b>	<b>Lecturer(s)</b>	<b>Year of Study</b>
1 <sup>st</sup> Cycle	Andreas Savva	1 <sup>st</sup>
<b>Mode of Delivery</b>	<b>Work Placement</b>	<b>Corequisites</b>
Face-to-face	N/A	COMP-116

### Course Objectives:

The main objectives of the course are to:

- introduce students to structured programming by means of the syntax and semantics of a structured high-level programming language.
- provide students a good working knowledge of a programming language. This includes programming constructs such as expressions, selection statements, loops, functions and arrays.
- provide practical experience in problem solving, coding, debugging, and testing.
- guide the student in order to develop good programming practices.
- obtain a foundation that will allow the student to pursue more advanced programming topics.

### Learning Outcomes:

After completion of the course students are expected to be able to:

- deal with the practicalities of writing a computer program.
- think and plan in a logical manner.
- apply a structured approach to problem solving.
- analyze and explain the behavior of simple programs involving the fundamental programming constructs.
- modify and expand short programs that use standard conditional and iterative controls structures and functions.

- design, implement, test and debug a program that uses each of the following fundamental programming constructs: Basic computation, Simple I/O, Standard conditional and iterative structures, Functions, Arrays.
- choose appropriate conditional and iteration constructs for a given programming task.
- apply the techniques of structured (functional) decomposition to break a program into smaller pieces.
- describe the mechanics of parameter passing (value and reference) and write programs with actual and formal parameters.

### **Course Content:**

1. Program design fundamentals
  - a. Problem solving
  - b. Program structure and basic programming concepts
2. Primitive data types and declarations
  - a. Input / Output
  - b. Constants, Variables, Numbers
  - c. Expressions, Arithmetic Statements, Standard functions
  - d. Formatted output
3. Decision statements
  - a. Boolean expressions
  - b. Relational operators
  - c. Decision Statements
4. Repetition statements
  - a. Pre-test loops
  - b. Post-test loops
5. Functions and scope rules
  - a. Parameter passing to functions (value and reference)
  - b. Function returning values
  - c. Scope and life-time of variables
6. Introduction to Arrays

### **Learning Activities and Teaching Methods:**

Lectures, In-Class Exercises, Computer Lab exercises

**Assessment Methods:**

Homework, Assignments, Mid-Term, Final Exam
---

**Required Textbooks / Readings:**

Title	Author(s)	Publisher	Year	ISBN
Introduction to Programming with C++, 3 <sup>rd</sup> ed.	Daniel Y. Liang	Pearson Education	2014	978-0-273-79324-3

**Recommended Textbooks / Readings:**

Title	Author(s)	Publisher	Year	ISBN
C++ Programming for the Absolute Beginner	Lee Mark, Henkemans Dirk	Course Technology	2009	978-1598638752