Course Syllabus

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>ECTS Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMP-111</td>
<td>Programming Principles I</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prerequisites</th>
<th>Department</th>
<th>Semester</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Computer Science</td>
<td>Fall, Spring</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type of Course</th>
<th>Field</th>
<th>Language of Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compulsory</td>
<td>Computer Science</td>
<td>English/Greek</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Level of Course</th>
<th>Lecturer(s)</th>
<th>Year of Study</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Cycle</td>
<td>Andreas Savva</td>
<td>1st</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mode of Delivery</th>
<th>Work Placement</th>
<th>Corequisites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face-to-face</td>
<td>N/A</td>
<td>COMP-112</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Title</th>
<th>Author(s)</th>
<th>Publisher</th>
<th>Year</th>
<th>ISBN</th>
</tr>
</thead>
</table>

Recommended Textbooks / Readings:

<table>
<thead>
<tr>
<th>Title</th>
<th>Author(s)</th>
<th>Publisher</th>
<th>Year</th>
<th>ISBN</th>
</tr>
</thead>
<tbody>
<tr>
<td>C++ Programming for the Absolute Beginner</td>
<td>Lee Mark, Henkemans Dirk</td>
<td>Course Technology</td>
<td>2009</td>
<td>978-1598638752</td>
</tr>
</tbody>
</table>