UNIVERSITY of NICOSIA

Course Syllabus

Course Code	Course Title	ECTS Credits
COMP-113	Programming Principles II	6
Prerequisites	**Department**	**Semester**
COMP-111	Computer Science	Spring, Fall
Type of Course	**Field**	**Language of Instruction**
Required	Computer Science	English
Level of Course	**Lecturer(s)**	**Year of Study**
1st Cycle	Prof. Athena Stassopoulou	1st
Mode of Delivery	**Work Placement**	**Corequisites**
Face to Face	N/A	None

Course Objectives:

The main objectives of the course are to:

- build on the knowledge already acquired in Programming I, by focusing on the more advanced concepts of procedural programming.
- enable the students to develop algorithmic thinking and problem solving.
- introduce dynamic memory allocation.
- provide practical experience in manipulating data strings, arrays, pointers and structures.
- provide the fundamentals of recursion.
- introduce students to I/O file stream and data files.
- introduce Object-Oriented Programming.

Learning Outcomes:

After completion of the course students are expected to be able to:
1. use built-in data structures to organize and process information.
2. develop algorithms and choose suitable data structures to produce solutions for complex problems using procedural programming.
3. develop working programs using the more advanced concepts such as pointers and arrays of structures.
4. develop programs using dynamic memory allocation.
5. describe and use recursion.
6. develop programs that use I/O file streams and data files.
7. apply the fundamentals of Object-Oriented Programming.

8. develop simple object-oriented programs.
9. develop applications consisting of multiple source files.

Course Content:

1. Revision (Loops and Functions).
2. Arrays (one dimensional, two dimensional), C-Strings.
3. Addresses and Pointers. Dynamic memory allocation, array names as pointers, pointer arithmetic, passing addresses, passing arrays, pointer arrays.
4. Data Structures: Structures, arrays of structures, structures as function arguments, passing a pointer, returning structures, dynamic data structure allocation, searching an array of structures.
5. Recursion. How the computation is performed by use of the stack, recursion versus iteration.
6. I/O File Streams and Data Files: I/O File Stream Objects and Methods, reading and writing character-based files, exceptions and file checking.
7. Introduction to Object Oriented Programming: Introduction to Objects and Classes, class functions.

Learning Activities and Teaching Methods:

Lectures, Lab Presentations, Lab Tutorials, Practical Exercises and Assignments.

Assessment Methods:

Mid-term exam, Projects, Assignments, Final Exam.

Required Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Introduction to C++. Programming and Data Structures, 5th ed.	Daniel Y. Liang	Pearson	2022	9780137391400

Recommended Textbooks / Readings:

Title	Author	Publisher	Year	ISBN
C++ How to Program, 2nd edition	Deitel & Deitel	Prentice Hall Inc.	1998	0-13-528910-6
The C++ Programming Language, 3rd edition,	Bjarne Stroustrup	Addison-Wesley	1997	0-201-88954-4
C++ Primer	J. LaJoie, S. B. Lippman & B. E. Moo	Addison-Wesley	2005	0-201-72148-1