UNIVERSITY of NICOSIA

Course Syllabus

Course Code	Course Title	ECTS Credits
ECE-113	Software Development Lab I	6
Prerequisites	**Department**	**Semester**
None	Computer Science	Fall, Spring
Type of Course	**Field**	**Language of Instruction**
Required	Computer Science	English/Greek
Level of Course	**Lecturer(s)**	**Year of Study**
1st Cycle	Dr Andreas Savva	1st
Mode of Delivery	**Work Placement**	**Corequisites**
Face-to-face	N/A	COMP-111

Course Objectives:

The main objectives of the course are to:

- introduce to the students good software development practices
- provide practical experience in developing software with appropriate comments and comment tags
- provide practical experience in developing readable, maintainable, robust, and secure source code
- provide practical experience in developing software which checks all function arguments and the function return argument
- provide practical experience in developing function/method tests and automatic test suites
- introduce tools/environments which automatically can run test suites
- introduce tools which automatically check the quality of the code
- introduce environments which provide code check-style.

Learning Outcomes:

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

- be proficient in developing high quality source code.
- describe what high quality source code is.
- demonstrate the ability to use tools to run automatic test suites.
- demonstrate the ability to use tools in order to test the quality and/or complexity of source code.

- be proficient in using development environments, which provides check-styles, and other tools for developing high quality source code.

Course Content:

1. Introduce the concept of developing high quality source code.
2. Develop source code with appropriate comments, comment blocks, and comment tags.
3. How to develop readable, maintainable, robust, and secure source code.
4. How to protect functions, methods, and the application by checking the state of incoming and/or outgoing arguments.
5. Use features and/or programming patterns of the programming language (like const, assert, safe-casting mechanisms) in order to increase quality of the source code.
6. Develop test cases and test suites to verify the correctness of the source code and apply automatic testing tools.
7. Use tools in order to check code complexity, dead-code, and other unwanted elements of the developed source code.
8. Introduction to code style and how to set up automatic check-style tools within the development environment to enforce/police the coding-style.

Learning Activities and Teaching Methods:

Lectures, lab presentations, lab tutorials, practical exercises, assignments

Assessment Methods:

Homework, Assignments, Mid-Term, Final Exam

Required Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Introduction to Programming with C++, 3rd 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