



Course Code OGEE-111	Course Title Programming for Engineers	ECTS Credits 8
Department Engineering	Semester Fall, Spring	Prerequisites None
Type of Course Required	Field Computer Science	Language of Instruction English
Level of Course 1 st Cycle	Year of Study 1 st	Lecturer(s) Dr Stelios Neophytou
Mode of Delivery Face-to-face	Work Placement N/A	Co-requisites None

Objectives of the Course:

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 should 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 Contents:

1. Program design fundamentals
 - a. Problem solving, Flow charts
 - 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
The course format is 3 h lectures and 1 h laboratory tutorial session per week.

Assessment Methods:

Homework, Assignments, Lab Reports, Mid-Term, Final Exam.

Required Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
Gary J Bronson	Program Development and Design Using C++	Thomson Course Technology	2006	0-619-21677-8

Recommended Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
Daniel Y. Liang	Introduction to Programming with C++	Pearson Education	2007	0-13-232049-5
Deitel & Deitel	C++ How to Program	Prentice Hall	2008	0-13-615250-7