

COURSE OUTLINE

(1) GENERAL

SCHOOL	SCHOOL OF SCIENCES		
ACADEMIC UNIT	DEPARTMENT OF MATHEMATICS		
LEVEL OF STUDIES	UNDERGRADUATE PROGRAM		
COURSE CODE		SEMESTER	C
COURSE TITLE	PROGRAMMING LANGUAGES		
INSTRUCTORS	Panagiotis Nastou (Lecture), Nikolaos Papaloukas and Christos Tsagaris (Lab sessions)		
INDEPENDENT TEACHING ACTIVITIES		WEEKLY TEACHING HOURS	CREDITS
		4	6
COURSE TYPE	Special background		
PREREQUISITE COURSES:	NO		
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	GREEK		
IS THE COURSE OFFERED TO ERASMUS STUDENTS	YES		
COURSE WEBSITE (URL)	http://www.math.aegean.gr/index.php/en/academics/undergraduate-programs		

(2) LEARNING OUTCOMES

Learning outcomes
<p>In this course the students are introduced to basic principles of programming languages. Initially, using the C programming language the students learn about structural and modular programming by using user defined functions, the preprocessor and standard libraries. They also learn about the definition of variables and constants and the use of operators as well. The mechanisms of managing memory and the class of variables that a compiler handles are also analyzed. The concept of pointers, arrays, structures are introduced, and the student learns how to use these in order to create more complex data structures. Files as an alternative way of data streaming to (input) or from (output) a program and communication mechanisms between programs and operating system as well are introduced. At the end of the course, the students are introduced to the concept of object and class of objects which are the basis of object oriented programming languages and to C++ as an object oriented version of C. Finally, they learn how to develop a program that solves a math problem using C++.</p> <p>At the end of this course, the student will be able to develop effective programs by using the C and C++ programming language. Moreover, they will be able to use the obtained foundations in comparing programs developed in different programming languages and in developing complex systems as well.</p>
General Competences
Working independently. Team working. Working in an interdisciplinary environment. Working in an international environment.

(3) SYLLABUS

- | |
|---|
| <ol style="list-style-type: none"> 1. Classes of High level programming languages. Structural Elements of programming languages. Compilers. Algorithmic Theory. Structured and Modular Programming and their applications on problem solving. 2. Introduction to the C programming language: Structure of a program in C. Standard Libraries. |
|---|

<p>Basic Data types (int, char, long, short, float, double). Variables and Constants (literals and symbolic). Arithmetic, logical-bit and shift operators. Relational and logical operators. Assignment and comma operators. Arithmetic and logical expressions. Priority of Operators and type conversion rules. Conditional Expressions and basic input/output functions.</p>	
<p>3. Flow Control commands (Conditional and loop commands). Examples of C programs. Functions and Procedures: definition and declaration. Local variables, typical parameters and arguments of a function. Parameter passing. Examples of C programs with functions. Recursion: design and coding of recursive functions for problem solving, performance of recursive functions and examples where the use of recursive functions leads to limited performance. Stack structure and its basic functions.</p>	
<p>4. Modular programming, global, static variables, register variables. Scope of Variables. The C preprocessor and the header files. Code Comments. Examples of Modular Programming. Basic Principles of memory organization. Memory Allocation per data type. Data Arrays: processing data of an array, passing an array as a parameter, assigning values to array elements. Strings.</p>	
<p>5. The notion of pointer, pointer arithmetic, pointers to single data type variables, pointers, pointers to arrays. Pointers and arrays. The functions as an argument. String manipulation functions of the standard library. The notion of dynamic memory allocation: heap, runtime memory allocation process, operating system and heap management, use and returning back the allocated memory and error handling. The use of parameters argc and argv of function main in the communication of a program with the operating system.</p>	
<p>6. Structures: Definition and Declaration. Structures and Arrays. Pointers to structures. Linked Lists: creation, search an element in an array and insert/delete nodes. Application: stack and queue implementation using linked lists.</p>	
<p>7. Input/Output mechanism in C: data streams, standard input/output data streams (stdin/stdout). Disk files: binary and text files, read/write/append functions, error handling during file operations.</p>	
<p>8. Introduction to Object-Oriented programming: Classes and Objects. Messages and Methods. Basic Types and reference types. Memory Management issues. Class Inheritance: Techniques.</p>	
<p>9. Fundamental elements of C++: Variables, basic types, operators and type conversion. Object creation, methods call and object conversion. Parameter passing. Arrays of objects. Flow control commands.</p>	
<p>10. Class and methods creation: Class implementation, Private methods and friend functions. Inline code.</p>	
<p>11. Input/Output mechanism in C++: Classes and methods for implementing input/output streams in C++ programs.</p>	
<p>12. Principles of program and algorithm design using object-oriented programming.</p>	
TEACHING MATERIAL DISTRIBUTION	The teaching material of the course is uniformly distributed during the semester.

(4) TEACHING and LEARNING METHODS - EVALUATION

DELIVERY	Face-to-face lectures	
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY	<ul style="list-style-type: none"> • Use of ICT in teaching • Communication with students via e-mail. • Uploading course material on moodle system. 	
TEACHING METHODS	Activity	Semester workload
	Lectures	39
	Laboratory Practice	13
	Independent study	98
	Course total (25 per ECTS)	150
COURSE COMMITMENTS	Attending course and lab sessions is not obligatory.	
STUDENT PERFORMANCE EVALUATION	Student's evaluation is done in Greek through a written examination which includes problem solving and laboratory	

	work. For students with disabilities, evaluation takes place via oral exams.
--	--

(5) ATTACHED BIBLIOGRAPHY

- | |
|--|
| <ol style="list-style-type: none">1. C++ for Mathematicians: An Introduction for Students and Professionals, Edward Scheinerman, Chapman & Hall/CRC.2. Tutor 's Electronic Notes on C Programming Language which are provided to students after they are registering in the course's site in moodle platform. |
|--|