

## ■ CE55 - Computer Programming with C

### GENERAL

<b>SCHOOL</b>	EXACT SCIENCES		
<b>DEPARTMENT</b>	MATHEMATICS		
<b>LEVEL OF STUDIES</b>	UNDERGRADUATE		
<b>COURSE CODE</b>	CE55	<b>SEMESTER</b>	E
<b>COURSE TITLE</b>	COMPUTER PROGRAMMING WITH C		
<b>INDEPENDENT TEACHING ACTIVITIES</b>	<b>WEEKLY TEACHING HOURS</b>	<b>ECTS</b>	
Lectures	4	5	
<b>COURSE TYPE</b>	Skills Development		
<b>PREREQUISITE COURSES</b>	Introduction to Programming		
<b>LANGUAGE OF TEACHING AND EXAMINATIONS</b>	Greek/English		
<b>THE COURSE IS OFFERED TO ERASMUS STUDENTS</b>	YES		
<b>COURSE WEBSITE (URL)</b>	<a href="http://eclass.uowm.gr/">http://eclass.uowm.gr/</a>		

### LEARNING OUTCOMES

<b>Learning Outcomes</b>
<p>The course aims to familiarize the students with the basic concepts of computer programming with the C language. Upon successful completion of the course, the student will be able to:</p> <ul style="list-style-type: none"> <li>• understand the basic structure of a program with C,</li> <li>• understand the concept of variables and different categories of data types,</li> <li>• become familiar with the properties and value ranges of different types of variables,</li> <li>• create new, complex data types,</li> </ul>

- become familiar with the selection structures and their variations,
- become familiar with repetition structures and their variations,
- understand the concept of functions, their gradient and their implementation (as sub-programs),
- become familiar with how to pass parameters to functions, and output values from functions via return value and parameters,
- know how to declare and use pointers to each type of data (ready or complex),
- understand the concept of stack and heap,
- create statically declared and dynamic arrays, by committing memory on the stack and on the heap, respectively,
- pass arrays to functions, either for input or output,
- use appropriate repetition structures and functions to solve basic mathematical problems,
- apply C programming in practice to solve complex problems,
- know introductory concepts of programming in C++,
- collaborate, where appropriate, with fellow students to create and present simple applications that demonstrate the concepts of each module.

### **General Competencies**

- Individual work.
- Team work.
- Project Planning and Management.
- Exercise criticism and self-criticism.

## **CONTENT OF THE COURSE**

The course introduces the fundamental concepts of computer programming with the C language. An industrial-scale compiler and development environment (IDE) are used to familiarize the students with professional tools. Basic concepts of basic variables, their types, capabilities and usage are covered. Furthermore, the creation of complex programmer-defined types is described in detail. Also, all selection and repetition structures are studied, with characteristic examples of conversion between them (depending on the category and the requirements). Memory management, both static and dynamic, is discussed in detail, along with pointers, dynamically allocating and freeing memory on/from the heap, and using it as one-dimensional or multi-dimensional, dynamically mutable arrays. Functions, input and output of values to/from them, and demonstration of them in common math problems are presented. These are applied to mathematical algorithms of various objects, such as e.g. numerical analysis, geometry, etc.

More specifically, the content of the course covers the following:

Section 1: The basic structure of a program in C. Introduction to the development environment of Visual Studio 2019. Introduction to the concept of variables of various types and classes. Limits and capabilities of ready-made data types. Operations with variables.

Section 2: Creating new, complex data types. Accessing the variable fields of complex

types. Location of fields and/or sub-fields in memory.

Section 3: Selection structures (if and switch), as well as a comparison between them. Extensive examples of their use.

Section 4: Repetition structures (for, while, do-while), as well as a comparison between them. A summary description of them, with the aim of making the best choice for the student, depending on each case. Convert between them, as well as referencing the basic iteration loop.

Section 5: Introduction to functions, using their parameters for input and output, and the optional return value.

Section 6: Static (stack) and dynamic memory (heap). Pointer declaration, dynamic memory allocation for ready or complex types.

Section 7: Statically and dynamically declared 1, 2, 3, 4 and 5 dimensional arrays. Passing arrays to functions for input and output.

Section 8: Use appropriate recurrence structures and functions to solve various basic mathematical problems. Practical application of C programming to solve complex problems.

Section 9: Demonstration of the use of the material of the previous sections in iterative methods (Conjugate Gradient and Jacobi-Accelerated Preconditioned Conjugate Gradient) of solving large-scale numerical systems. Solving integrals with the trapezium method, matrix operations, problems limited by memory or processor. Accurate timing of the above. Examples of parallel code in math problems.

Section 10: Introduction to basic concepts of object oriented programming with C++.

Section 11: Standard functions.

## TEACHING AND LEARNING METHODS - EVALUATION

<b>TEACHING METHOD</b>	In the classroom.	
<b>USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY</b>	Software MATLAB. e-Lectures. Use of e-class. Communication through face-to-face discussions and e-mails.	
<b>TEACHING ORGANIZATION</b>	<b>Activity</b>	<b>Semester Workload</b>
	Lectures	52 hours
	Projects	18 hours
	Individual Study	55 hours
	Course Total (25 hours per ECTS)	125 hours

<b>STUDENT EVALUATION</b>	<p>Written final exam (100%) in the Greek language, which includes:</p> <ul style="list-style-type: none"> <li>- Short answer questions</li> <li>- Problem solving</li> <li>- Optional exercises and tasks, individual or group</li> <li>- Comparative assessment of theory.</li> </ul> <p>During the semester, the students are given individual assignments or group exercises and assignments, as well as a larger optional group assignment covering several subject areas simultaneously.</p>
---------------------------	--

### **RECOMMENDED BIBLIOGRAPHY**

1. Cheng H. (2012) C for scientists and engineers. 1st Edition. Tziola Publications. (Greek)
2. Tan H and D' Orazio T. (2000) C for engineers. 1st Edition. Tziola Publications. (Greek)
3. Tselikis GS and Tselikas ND. (2016) C: From theory to practice. 3rd Edition. Publisher: Tselikis Georgios. (Greek)
4. Hanly Jeri R. and Koffman Elliot B. (2021) Principles and Techniques of Programming with the C Language. 1st Edition. Kritiki Publications S.A. (Greek)
5. Hatzigiannakis NM. (2017) The C language in depth. 5th Edition. Kleidaritmos S.A. Publications. (Greek)
6. Paul Deitel, Harvey Deitel. (2014) A Programmer's Guide to C. 1st Edition. Ch Giourda & Co Publications S.A. (Greek)
7. Abbey Deitel, Harvey Deitel. (2014) C Programming. 7th Edition. Ch Giourda & Co Publications SA. (Greek)
8. Karolidis Dimitrios A. (2021) Learn C easily. 2nd Edition. Pinelopi Xarhakou Publications. (Greek)