

## COURSE OUTLINE

### GENERAL

<b>SCHOOL</b>	School of Science		
<b>ACADEMIC UNIT</b>	Department of Mathematics		
<b>LEVEL OF STUDIES</b>	Undergraduate		
<b>COURSE CODE</b>	<b>MAY343</b>	<b>SEMESTER</b>	<b>3</b>
<b>COURSE TITLE</b>	Introduction to Programming		
<b>INDEPENDENT TEACHING ACTIVITIES</b> <i>if credits are awarded for separate components of the course, e.g. lectures, laboratory exercises, etc. If the credits are awarded for the whole of the course, give the weekly teaching hours and the total credits</i>	<b>WEEKLY TEACHING HOURS</b>	<b>CREDITS</b>	
lectures, laboratory exercises, tutorials, quiz	5	5	
<i>Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (d).</i>			
<b>COURSE TYPE</b> <i>general background, special background, specialised general knowledge, skills development</i>	General background		
<b>PREREQUISITE COURSES:</b>			
<b>LANGUAGE OF INSTRUCTION and EXAMINATIONS:</b>	Greek		
<b>IS THE COURSE OFFERED TO ERASMUS STUDENTS</b>	Yes		
<b>COURSE WEBSITE (URL)</b>	<a href="http://www.cs.uoi.gr/~charis/c343/">http://www.cs.uoi.gr/~charis/c343/</a>		

### LEARNING OUTCOMES

<p><b>Learning outcomes</b>  <i>The course learning outcomes, specific knowledge, skills and competences of an appropriate level, which the students will acquire with the successful completion of the course are described.</i></p> <p><i>Consult Appendix A</i></p> <ul style="list-style-type: none"> <li>• <i>Description of the level of learning outcomes for each qualifications cycle, according to the Qualifications Framework of the European Higher Education Area</i></li> <li>• <i>Descriptors for Levels 6, 7 &amp; 8 of the European Qualifications Framework for Lifelong Learning and Appendix B</i></li> <li>• <i>Guidelines for writing Learning Outcomes</i></li> </ul>
<p>This course aims at introducing to students the philosophy of programming and at giving them the ability to implement algorithms in C/C++.</p> <p>After successfully passing this course the students will be able to:</p> <ul style="list-style-type: none"> <li>• Write simple or complex programs</li> </ul>

- Verify the correctness and appropriateness of a given program
- Debug programs
- Understand basic programming concepts, structures and techniques
- Use arrays, strings, and functions
- Conduct simple and complex arithmetic computations via programming
- Use control flow constructs, conditions, decision structures and loops
- Structure their programs with the help of iterative and recursive functions
- Program basic operations on data, such as searching and sorting

### General Competences

*Taking into consideration the general competences that the degree-holder must acquire (as these appear in the Diploma Supplement and appear below), at which of the following does the course aim?*

*Search for, analysis and synthesis of data and information, with the use of the necessary technology*

*Adapting to new situations*

*Decision-making*

*Working independently*

*Team work*

*Working in an international environment*

*Working in an interdisciplinary environment*

*Production of new research ideas*

*Project planning and management*

*Respect for difference and multiculturalism*

*Respect for the natural environment*

*Showing social, professional and ethical*

*responsibility and sensitivity to gender issues*

*Criticism and self-criticism*

*Production of free, creative and inductive thinking*

*Others*

- Search for, analysis and synthesis of data and information, with the use of the necessary technology
- Working independently
- Team work
- Project planning and management

### SYLLABUS

- i) Introduction to programming and binary representation
- ii) Input/Output, data structures and variables
- iii) Preprocessing, numerical, boolean and logical operators
- iv) Flow control: if/else, switch, for, while, do-while
- v) Structuring, locality of parameters, pass by value/reference, variable scope, recursive functions, program stack.
- vi) Arrays
- vii) Searching and sorting data
- viii) String operations
- ix) Type and data structures and file processing

### TEACHING and LEARNING METHODS - EVALUATION

<b>DELIVERY</b> <i>Face-to-face, Distance learning, etc.</i>	Lectures, labs session
<b>USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY</b> <i>Use of ICT in teaching, laboratory education, communication with students</i>	<ul style="list-style-type: none"> <li>• Use of projector and interactive board during lectures.</li> <li>• Use of computer for demonstration of programming.</li> <li>• Use of computers in laboratories for development and testing of programs.</li> <li>• Course website maintenance. Announcements</li> </ul>

	<p>and posting of teaching material (lecture slides and notes, programs).</p> <ul style="list-style-type: none"> <li>Announcement of assessment marks via the ecourse platform by UOI.</li> </ul>																				
<p><b>TEACHING METHODS</b>  <i>The manner and methods of teaching are described in detail. Lectures, seminars, laboratory practice, fieldwork, study and analysis of bibliography, tutorials, placements, clinical practice, art workshop, interactive teaching, educational visits, project, essay writing, artistic creativity, etc. The student's study hours for each learning activity are given as well as the hours of non-directed study according to the principles of the ECTS</i></p>	<table border="1"> <thead> <tr> <th><i>Activity</i></th> <th><i>Semester workload</i></th> </tr> </thead> <tbody> <tr> <td>Lectures</td> <td>75</td> </tr> <tr> <td>Laboratory practice</td> <td>25</td> </tr> <tr> <td>Tutorials</td> <td>25</td> </tr> <tr> <td></td> <td></td> </tr> <tr> <td></td> <td></td> </tr> <tr> <td></td> <td></td> </tr> <tr> <td></td> <td></td> </tr> <tr> <td></td> <td></td> </tr> <tr> <td>Course total</td> <td><b>125</b></td> </tr> </tbody> </table>	<i>Activity</i>	<i>Semester workload</i>	Lectures	75	Laboratory practice	25	Tutorials	25											Course total	<b>125</b>
	<i>Activity</i>	<i>Semester workload</i>																			
	Lectures	75																			
	Laboratory practice	25																			
	Tutorials	25																			
Course total	<b>125</b>																				
<p><b>STUDENT PERFORMANCE EVALUATION</b>  <i>Description of the evaluation procedure  Language of evaluation, methods of evaluation, summative or conclusive, multiple choice questionnaires, short-answer questions, open-ended questions, problem solving, written work, essay/report, oral examination, public presentation, laboratory work, clinical examination of patient, art interpretation, other  Specifically-defined evaluation criteria are given, and if and where they are accessible to students.</i></p>	<ul style="list-style-type: none"> <li>Final written examination (80%) <ul style="list-style-type: none"> <li>Multiple choice questions</li> <li>Develop programs</li> </ul> </li> <li>Laboratory exercises (20%) <ul style="list-style-type: none"> <li>Multiple choice questions</li> <li>Develop programs</li> </ul> </li> </ul>																				

#### ATTACHED BIBLIOGRAPHY

- Suggested bibliography:

- Related academic journals:

[1] W. Savitch, Πλήρης C++, Εκδόσεις Τζιόλα, 2011. Κωδικός Ευδ: 18548892

[2] H. Deitel and P. Deitel, C++ Προγραμματισμός 6η Έκδοση, Εκδόσεις Μ. Γκιούρδας, 2013. Κωδικός Ευδ: 12536819

[3] L. Jesse, Πλήρες εγχειρίδιο της C++, Εκδόσεις Α. Γκιούρδα, 2006. Κωδικός Ευδ: 12374

[4] Ν. Χατζηγιαννάκης, Η γλώσσα C++ σε βάθος, Εκδόσεις Κλειδάριθμος, 2008. Κωδικός Ευδ: 13761