

COURSE OUTLINE

(1) GENERAL

SCHOOL	SOCIAL SCIENCES		
ACADEMIC UNIT	DEPARTMENT OF CULTURAL TECHNOLOGY AND COMMUNICATION		
LEVEL OF STUDIES	UNDERGRADUATE		
COURSE CODE	PLR 110	SEMESTER	3 rd
COURSE TITLE	OBJECT-ORIENTED PROGRAMMING I		
INDEPENDENT TEACHING ACTIVITIES <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>		WEEKLY TEACHING HOURS	CREDITS
Lectures and Laboratories		4	6
<i>Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (d).</i>			
COURSE TYPE <i>general background, special background, specialised general knowledge, skills development</i>	Core Course/Skills Development		
PREREQUISITE COURSES:	Introduction to Programming		
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	Greek		
IS THE COURSE OFFERED TO ERASMUS STUDENTS	Yes		
COURSE WEBSITE (URL)	https://eclass.aegean.gr/courses/131371/		

(2) LEARNING OUTCOMES

<p>Learning outcomes</p> <p><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"> ● <i>Description of the level of learning outcomes for each qualifications cycle, according to the Qualifications Framework of the European Higher Education Area</i> ● <i>Descriptors for Levels 6, 7 & 8 of the European Qualifications Framework for Lifelong Learning and Appendix B</i> ● <i>Guidelines for writing Learning Outcomes</i>
<p>After the completion of the specific course students will be able to:</p> <ul style="list-style-type: none"> ● Know the basic principles of object oriented programming and how to implement them. ● Understand the importance of object oriented application design to solve problems. ● Understand the syntax and how the programming language entities are used. ● Create programs with the C++ programming language following the principles of object-oriented programming. ● Understand easily and quickly other object oriented programming languages like Java and C#.
<p>General Competences</p> <p><i>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?</i></p> <p><i>Search for, analysis and synthesis of data and information, with the use of the necessary technology</i> <i>Project planning and management</i> <i>Respect for difference and multiculturalism</i></p>

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

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 in an interdisciplinary environment
- Production of free, creative and inductive thinking
- Transfer of know-how in other environments

(3) SYLLABUS

This course concerns the introduction to object oriented programming. The transition from procedural to object-oriented programming is a challenge and the objective of this course. C++ language was chosen which, as an extension of C, combines these two philosophies of programming and makes the transition from one programming philosophy to another, smoother and less painful. All the concepts of object oriented programming are fully presented and explained using the tools of the language. Particular attention is given to the design of object-oriented programs and not only the use of object-oriented tools. The knowledge gained from the monitoring of the course make it easy to transition to more "purebred" object-oriented programming languages such as Java and C #.

Lecture schedule	
No	Lecture
1	Introduction to Object Oriented Programming - Differences from Procedural. Basic concepts.
2	The cin and cout objects, Our first program, Variables, operators and representations, the ++ and - operators, The if statement
3	Repeat structures, while, do..while and for, Break and continue commands - Variable address, sizeof and & operators - String handling - Defining and calling functions
4	Function overloading, Preset parameter values, Arrays, Introduction to classes.
5	Introduction to Classes, Member Variables and Member Functions, Defining Classes and Objects, The String Class
6	Constructors and destructors, Method overloading, Object arrays, Passing arrays to functions
7	Object Returning Functions, UML, Pointers to objects, Nested Classes
8	This pointer, Operator overloading
9	Formatted output - operators and methods of cout and cin objects, Static class members
10	Protected members - Inheritance - Correlation of classes. Derived and basic classes.
11	Stream classes and objects, File handling.
12	Class prototypes and Exceptions
13	Recap

(4) TEACHING and LEARNING METHODS - EVALUATION

<p style="text-align: center;">DELIVERY <i>Face-to-face, Distance learning, etc.</i></p>	Face-to-face																									
<p style="text-align: center;">USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY <i>Use of ICT in teaching, laboratory education, communication with students</i></p>	Use of open source software for laboratory education. Use ICT in teaching and communication with students.																									
<p style="text-align: center;">TEACHING METHODS <i>The manner and methods of teaching are described in detail.</i> <i>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.</i></p> <p><i>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" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;"><i>Activity</i></th> <th style="text-align: center;"><i>Semester workload</i></th> </tr> </thead> <tbody> <tr> <td>Lectures</td> <td style="text-align: center;">13 *2 hours = 26 hours</td> </tr> <tr> <td>Lectures' study</td> <td style="text-align: center;">13*3 hours = 39 hours</td> </tr> <tr> <td>Laboratory Practice</td> <td style="text-align: center;">13*2 hours = 26 hours</td> </tr> <tr> <td>Laboratory Preparation</td> <td style="text-align: center;">13*3 hours = 39 hours</td> </tr> <tr> <td>Tutorials</td> <td style="text-align: center;">13*2 hours = 26 hours</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 style="text-align: center;">156 hours</td> </tr> </tbody> </table>		<i>Activity</i>	<i>Semester workload</i>	Lectures	13 *2 hours = 26 hours	Lectures' study	13*3 hours = 39 hours	Laboratory Practice	13*2 hours = 26 hours	Laboratory Preparation	13*3 hours = 39 hours	Tutorials	13*2 hours = 26 hours											Course total	156 hours
<i>Activity</i>	<i>Semester workload</i>																									
Lectures	13 *2 hours = 26 hours																									
Lectures' study	13*3 hours = 39 hours																									
Laboratory Practice	13*2 hours = 26 hours																									
Laboratory Preparation	13*3 hours = 39 hours																									
Tutorials	13*2 hours = 26 hours																									
Course total	156 hours																									
<p style="text-align: center;">STUDENT PERFORMANCE EVALUATION <i>Description of the evaluation procedure</i></p> <p><i>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</i></p> <p><i>Specifically-defined evaluation criteria are given, and if and where they are accessible to students.</i></p>	<p>The final exam is done with a written exam at the end of the semester and with the implementation of semester assignments. Students are examined with open notes.</p> <p>The evaluation criteria become known during the initial-introductory lecture and are posted throughout the semester on the website for storing the course content (eclass.aegean.gr). The evaluation of students is based on the grade of the final written examination in all the taught material in a percentage of 70% and the assignments that receive 30% of the grade.</p>																									

(5) ATTACHED BIBLIOGRAPHY

- Suggested bibliography:

- Η ΓΛΩΣΣΑ C++ ΣΕ ΒΑΘΟΣ, ΧΑΤΖΗΓΙΑΝΝΑΚΗΣ Μ. ΝΙΚΟΣ, Έτος Έκδοσης: 2014, Εκδότης: Κλειδάριθμος, Αριθμός σελίδων: 976, Κωδικός ISBN: 978-960-461-620-6
- Αντικειμενοστρεφής προγραμματισμός με τη C++, Lafore, Robert, Έτος Έκδοσης: 2006, Εκδότης: Κλειδάριθμος, Αριθμός σελίδων: 1040, Κωδικός ISBN: 9602099046
- Ivor Horton, "ANSI C++ The Complete Language", Apress, 2004.
- Anthony Sintes, "Object Oriented Programming", Sams, 2002.
- Harvey M. Deitel, Paul J. Deitel, "C++: How to Program", Prentice Hall, 2000.
- Nell Dale, "Programming and Problem Solving with C++", Jones and Bartlett Publishers, 2001.
- Andrew Haigh, "Object-oriented Analysis and Design", McGraw-Hill, 2001.
- The C++ Resources Network, <http://www.cplusplus.com>

- Related academic journals:

- Programming and Computer Software, Springer
- New Generation Computing, Springer

