COURSE OUTLINE

1. GENERAL

SCHOOL	NATURAL SCIENCES		
DEPARTMENT	MATHEMATICS		
LEVEL OF COURSE	UNDERGRADUATE		
COURSE CODE	MAT_IC232	**SEMESTER OF STUDIES**	4th
COURSE TITLE	OBJECT-ORIENTED PROGRAMMING USING C++		

INDEPENDENT TEACHING ACTIVITIES *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*	**TEACHING HOURS PER WEEK**	**ECTS CREDITS**
Lectures and Laboratory exercises	4	6
Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (d).		

COURSE TYPE *general background, special background, specialised general knowledge, skills development*	Compulsory course for the specialization *Informatics and Computational Mathematics* Elective course for each of the other specializations
PREREQUISITE COURSES:	Recommended prerequisite knowledge: INTRODUCTION TO COMPUTERS AND PROGRAMMING WITH FORTAN, PROGRAMMING WITH PYTHON
TEACHING AND ASSESSMENT LANGUAGE:	Greek
THE COURSE IS OFFERED TO ERASMUS STUDENTS	Yes
COURSE WEBPAGE (URL)	https://eclass.upatras.gr/courses/MATH1060/

2. LEARNING OUTCOMES

Learning outcomes

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.

Consult Appendix A

- *Description of the level of learning outcomes for each qualifications cycle, according to the Qualifications Framework of the European Higher Education Area*
- *Descriptors for Levels 6, 7 & 8 of the European Qualifications Framework for Lifelong Learning*

and Appendix B

- *Guidelines for writing Learning Outcomes*

By successfully attending this course the student will be able to use the C++ programming language and the fundamental techniques of the Object-Oriented Programming for the solution of problems. More specifically, at the end of the course, the student will have developed the following skills:

- To use the preprocessing directives, the fundamental and composite data types, the expressions, the commands, the pointers, the references and the functions of C++. These are the basic tools of this language.
- To utilize the libraries of C++; to realize the properties of this language regarding the scope and visibility of the elements that compose the program.
- To apply the procedure orriented part of C++ for implementing basic algorithms.
- To design classes in C++, handle their objects and apply the capabilities of encapsulation, hiding, inherritance, dynamic binding and template usage that are provided by this language.

He/she will also be able to use all these in order to design and implement programs for the solution of elementary mathematical or other kinds of problems.

General Abilities

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, analyze and synthesize data and information, using the necessary technologies.
- Autonomous work.

3. COURSE CONTENT

<u>Lectures</u>

Introduction to C++. The translation process of a program in C++, preprocessing directives, fundamental and composite data types, type conversion, constants and variables, operators and expressions, declarations and statements input and output, functions, pointers and references, arrays, structures and unions.

Object-Oriented Programming in C++. Abstraction, classes and class members, construction, and usage of class objects, accessibility of class members, inheritance, function and operator overloading, virtual member functions and classes, abstract classes, function and class templates.

<u>Laboratory exercises</u>

Familiarization with the subjects mentioned above. Application of C++ (Unix operating system) in problem solving.

4. TEACHING AND LEARNING METHODS - ASSESSMENT

<table>
<tr><td><b>TEACHING METHOD</b>
<i>Face-to-face, Distance learning, etc..</i></td><td>Lectures (face to face)</td></tr>
<tr><td><b>USE OF INFORMATION AND COMMUNICATION TECHNOLOGIES</b>
<i>Use of ICT in teaching, laboratory education, communication with students</i></td><td>The laboratory exercises and the corresponding exams are implemented in C++ under Unix operating system. The learning procedure is supported by the <i>eClass</i> platform of University of Patras. The students are encouraged to use e-mail and Skype in order to communicate with their teachers</td></tr>
<tr><td><b>TEACHING ORGANIZATION</b>
<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>
<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></td><td>
<table>
<tr><th><i>Activity</i></th><th><i>Semester workload</i></th></tr>
<tr><td>Lectures</td><td>26</td></tr>
<tr><td>Laboratory exercises</td><td>26</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Study (unsupervised)</td><td>39</td></tr>
<tr><td>Solving problems by using programming (semi-supervised)</td><td>52</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Laboratory examination</td><td>4</td></tr>
<tr><td>Final examination</td><td>3</td></tr>
<tr><td></td><td></td></tr>
<tr><td><b><i>Total number of hours for the Course (25 hours of work-load per ECTS credit)</i></b></td><td><b><i>150</i></b></td></tr>
</table>
</td></tr>
<tr><td><b>STUDENT ASSESSEMNT</b>
<i>Description of the evaluation procedure</i>
<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>
<i>Specifically-defined evaluation criteria are given, and if and where they are accessible to students.</i></td><td><b>Assessment Language:</b> Greek
<b>Assessment Language for Erasmus students:</b> English
<b>Assessment methods:</b>
✓ Laboratory examination (twice)
✓ Final examination
Minimum passing grade: 5
Maximum passing grade: 10</td></tr>
</table>

5. RECOMMENDED LITERATURE

(*in Greek*)

- Ράγγος Όμηρος. *Γλώσσες Προγραμματισμού Ι*. Σημειώσεις μαθήματος, 2015.
- Schildt Herbert. *C++ βήμα προς βήμα*. Εκδόσεις Γκιούρδας, 2005.
- Eckel Bruce. *Τρόπος σκέψης σε C++. Τόμος Ι και ΙΙ.* Εκδόσεις Γκιούρδας, 2009.

(*in English*)

- Schildt Herbert. *C++: A Beginner's Guide*. 3rd ed., McGraw-Hill, 2005.
- Eckel Bruce. *Thinking in C++. Vol. I and 2*. 2nd ed., Prentice Hall, 2000.