COURSE OUTLINE

(1) GENERAL

SCHOOL	SOCIAL SCIENCES		
ACADEMIC UNIT	DEPARTMENT OF CULTURAL TECHNOLOGY AND COMMUNICATION		
LEVEL OF STUDIES	UNDERGRADUATE		
COURSE CODE	3PLR 115	**SEMESTER**	4^{o}
COURSE TITLE	OBJECT-ORIENTED PROGRAMMING I		

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*	**WEEKLY TEACHING HOURS**	**CREDITS**
Lectures	2	
Laboratories	2	
The organization of teaching and the teaching methods used are described in detail at (d).	4	6

COURSE TYPE *General background, Special background, specialized general Knowledge, skills development*	Core Course/Special Background
PREREQUISITE COURSES	Object-Oriented Programming I
LANGUAGE OF INSTRUCTIONS and EXAMINATIONS	Greek
IS THE COURSE OFFERED TO ERASMUS STUDENTS	Yes
COURSE WEBSITE (URL)	https://eclass.aegean.gr/courses/131275/

(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 learning outcomes for each qualification cycle, according to the Qualifications Framework of the European Higher Education Area*
- *Descriptions for level 6, 7 & 8 of the European Qualification Framework for Lifelong Learning and Appendix B*
- *Guidelines for writing Learning Outcomes*

At the end of this course, the students will be able to:

- Be aware of the basic principles of Object Oriented Programming (OOP) and the JAVA programming language.
- Understand the importance of OO design using class objects and methods.
- Describe polymorphism and inheritance to create specialized classes.
- Know how to handle exceptions to resolve errors that might occur from the exceptions.
- Implement programs/applications applying the principles of the OOP.
- Generate executable programs/applications using web-based open source software.
- Communicate efficiently their knowledge, which is acquired from the lectures, to colleagues to establish fruitful co-operations for creating cultural informatics applications.
- Enrich knowledge in JAVA through practical examples and programming code.

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	*Project planning and management*
Adapting to new situations	*Respect for difference and multiculturalism*
Decision-making	*Respect for the natural environment*
Working independently	*Showing social, professional and ethical responsibility and sensitivity to gender issues*
Team work	*Criticism and self-criticism*
Working in an international environment	*Production of free, creative and inductive thinking*
Working in an interdisciplinary environment	……
Production of new research ideas	*Others…*
	…….

- Search for, analysis and synthesis of data and information, with the use of the necessary technology
- Production of free, creative and inductive thinking
- Transfer of know-how in other environments
- Practice Critical Thinking
- Mid-semester test

(3) SYLLABUS

This course is an extension of the "Object-Oriented Programming (OOP) I" course and concentrates on the JAVA programming language. Students exploit the capabilities of JAVA language to implement programs and applications with the use of ready-based libraries. JAVA language is chosen due to its popularity among all other languages on the application development of industrial, commercial, or any other activity. The basic course aim is the students' preparation and enriching knowledge on designing, implementing and developing applications with the use of web-based open source software. Special attention is given on the smooth transition of the "Object-Oriented Programming I" course of the 3 semester to the specific "Object-Oriented Programming II" course.

Lectures

1. Introduction – Course structure
2. From C++ to Java - Introduction to Object-oriented programming and specifically to JAVA programming language – Basic commands, variables, operators
3. Logical expressions – Decision (control) structure, enumeration – While loop For loop – break and continue commands
4. Classes and methods – Object creation
5. Constructors and ways to create objects
6. Arrays and parameters – Search and sort method – Multi-dimensional arrays
7. Inheritance and polymorphism – Super classes and subclasses
8. Private instance variables or attribute
9. Exceptions
10. Dynamic data structures
11. File Input and Output
12. Python and object oriented approach I
13. Python and object oriented approach II

(4) TEACHING and LEARNING METHODS - EVALUATION

<table>
<tr><td>DELIVERY
Face-to-face, Distance learning, etc.</td><td>Face-to-face</td></tr>
<tr><td>USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY
Use of ICT in teaching, laboratory education, communication with students</td><td>Use web-based open source software for laboratory education.
Use ICT in teaching and communication with students</td></tr>
<tr><td>TEACHING METHODS
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</td><td>
<table>
<tr><th>Activity</th><th>Semester workload</th></tr>
<tr><td>Lectures</td><td>13 *2 hours = 26 hours</td></tr>
<tr><td>Lectures' study</td><td>13*4 hours = 52 hours</td></tr>
<tr><td>Laboratory practice</td><td>13*2 hours = 26 hours</td></tr>
<tr><td>Laboratory preparation and semester assignment</td><td>36 hours</td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td>Course total</td><td>140 hours</td></tr>
</table>
</td></tr>
<tr><td>STUDENT PERFORMANCE EVALUATION
Description of the evaluation procedure</td><td>The evaluation of students' performance is conducted at the end of the semester with exams and with a mid-semester test. Students may use their books or notes from the lessons and the laboratory exercises (open book exams).</td></tr>
</table>

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.*	The evaluation criteria are announced during the first introductory lesson and they can be found at the storage content in the course's area in the University e-class platform (eclass.aegean.gr). The file with the first lesson contains all the information. The students' performance evaluation is based on the grade of the final exam with a weighted percentage of 70% (grade * 70%) and on the mid-semester test with a weighted percentage of 30% (grade * 30%). The mid-semester test is mandatory.

(5) ATTACHED BIBLIOGRAPHY

- Suggested bibliography:

- Java: An Introduction to Problem Solving and Programming (7th Edition), Walter Savitch
- JAVA Programming, (7th Edition), Joyce Farrell,
- Head First Java (2nd Edition), Kathy Sierra & Bert Bates

-Related academic bibliography

- Intro to Java Programming, Comprehensive Version (10th Edition), Y. Daniel Liang
- Effective Java (3rd Edition), Joshua Bloch