COURSE OUTLINE

(1) GENERAL

SCHOOL	SOCIAL SCIENCES		
ACADEMIC UNIT	DEPARTMENT OF CULTURAL TECHNOLOGY AND COMMUNICATION		
LEVEL OF STUDIES	UNDERGRADUATE		
COURSE CODE	PLR 100	SEMESTER	1st
COURSE TITLE	INTRODUCTION TO PROGRAMMING		

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 and Laboratories	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*	Core Course/General Background/Skills Development
PREREQUISITE COURSES:	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	Greek
IS THE COURSE OFFERED TO ERASMUS STUDENTS	Yes
COURSE WEBSITE (URL)	https://eclass.aegean.gr/courses/131184/

(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*

After the completion of the specific course students will be able to:

- Gain knowledge on the basic IT domains like software, hardware, networks, numeral systems
- Define and explain the basic principles of procedural, structured programming and the ways for applying them to real problems.
- Understand the importance of algorithmic logic and how it contributes on solving simple and complex algorithmic problems.
- Understand the syntactical rules of Python commands and their functionality.
- Understand and use Python environments and also to develop, debug and execute Python programs.
- Create Python programs by applying the principles of procedural programming to implement solutions to algorithmic problems.

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

(3) SYLLABUS

The course covers the teaching of IT and programming initially though a number of introductory lectures on Hardware and Software, numeral systems, networks and the Internet. The course continues to cover basic programming areas like general principles for program design, algorithms, flow charts, techniques for designing algorithms and solving algorithmic problems. Finally the last set of lectures focuses on the Introduction to Programming through the Python programming language by covering the aspects of Program elements: variables, constants, expressions, basic data types, operators as well as Data Input/Output commands, Decision control commands and Repeat structures.

Lectures	
1.	Introduction to Hardware: The PC parts
2.	Introduction to Software: The Operating System
3.	Introduction to Numeral Systems: The binary systems and transformations to other systems
4.	Introduction to Networks: Types and characteristics of networks
5.	Introduction to Internet: Architecture, useful tools and applications
6.	Introduction to basic principles of structured programming
7.	Introduction to Algorithms and Python. Python Programming environment
8.	Variables and Data types
9.	Numerical and Alphanumerical operators
10.	Simple Structure
11.	Conditions
12.	While and For Loops
13.	Revision

(4) TEACHING and LEARNING METHODS - EVALUATION

DELIVERY *Face-to-face, Distance learning, etc.*	Face-to-face
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY *Use of ICT in teaching, laboratory education, communication with students*	Use of open source software for laboratory education. Use ICT in teaching and communication with students.
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*	
STUDENT PERFORMANCE EVALUATION *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.*	The final evaluation is accomplished with a written exam at the end of the semester. Students are examined with open notes. Students are familiar with the evaluation criteria during the initial course lecture at the beginning of the semester and are stored throughout the semester in the course's area in eclass (eclass.aegean.gr).

Activity	*Semester workload*
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

(5) ATTACHED BIBLIOGRAPHY

- Suggested bibliography:

- Matthes Eric, Η γλώσσα προγραμματισμού PYTHON, ΔΙΣΙΓΜΑ, 2020, Θεσσαλονίκη, ISBN: 978-618-202-003-6
- Harvey Deitel, Paul Deitel, Python για Προγραμματιστές, Γκιούρδας, 2020, Αθήνα, ISBN: 978-960-512-7183
- Στράτος Καλαφατούδης, Γεώργιος Σταμούλης, Προγραμματισμός με την Python, Εκδόσεις Νέων Τεχνολογιών, 2018, Αθήνα, ISBN: 978-960-578-040-1
- Αριστείδης Σ. Μπούρας, Γιάννης Θ. Κάππος, Python3: Αλγοριθμική και Προγραμματισμός, Κλειδάριθμος, 2020, Αθήνα, ISBN: 978-960-645-087-7

- Related academic journals:

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