COURSE OUTLINE

1 GENERAL

SCHOOL	School of Sciences and Technology		
PROGRAMME	INFORMATICS		
LEVEL OF COURSE	UNDERGRADUATE		
COURSE CODE	ΠΛΗ-ΠΡΟ	**YEAR**	1st
COURSE TITLE	PROGRAMMING LABORATORY		
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**
Workload x 32 weeks		5	6 ECTS
COURSE TYPE *general background, special background, specialised general knowledge, skills development*	Background, skills development		
PREREQUISITE COURSES:	-		
TEACHING AND ASSESSMENT LANGUAGE:	Greek		
THE COURSE IS OFFERED TO ERASMUS STUDENTS	No (becauce of one year duration)		
COURSE WEBPAGE (URL)	https://www.eap.gr/education/undergraduate/computer-science/topics/ It has its own space in the digital education space of Hellenic Open University (http://study.eap.gr), with authorized access (with password) for students and teaching staff.		

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*

a) Upon successful completion of the laboratory, the students will know:
- the structure of a program in Python programming language
- the concepts of logical and syntactic errors, as well as debugging process
- the operation of decision commands in the Python programming language
- the logical operators and logical operations of the Python programming language
- the basic data types of the Python programming language
- the functionality of data structures: list, stack, queue, dictionary and tuples
- the basic concepts of object-oriented programming
- the basic concepts of event driven programming

b) Upon successful completion of the laboratory, students will know:
- the differences between iteration loops in python programming language (for, while)

- how to declare functions in the python programming language, and how to return a value (or not)
- the concept of the recursive functions, as well as the advantages / disadvantages over the iterative functions (with loop)
- the concept of modular programming
- the difference between mandatory and optional function parameters
- string management
- the concept of object-oriented programming using classes and objects
- the concept of inheritance and polymorphism
- the methodology of event driven programming

c) After the successful completion of the laboratory, the students will be able to:
- describe their solutions in computational steps in the Python programming language
- use modern data structures such as tuples, sets, sequences, dictionaries and lists.
- use and modify existing programs in Python (scripts and functions).
- use Python programming environments.
- use Python csv, json, pickle libraries for file management
- use Python's os, os.path libraries that allow interaction with operating system, the creation and deletion of folders, and display their contents.
- use the Python tkinter library for graphical interface design

- use the Python pandas library for data management and analysis
- use Python's NumPy library for scientific calculations with the available implementations of N-dimensional arrays and functions for linear algebra problems
- use the python matplotlib library to create high quality graphical representations

- debug programs that display syntax or logical errors

d) After the successful completion of the laboratory, the students will be able to:
- select the appropriate data structures and variables for the basic functions of their programs
- implement programs that use the appropriate data structures (lists, stacks, queues, dictionaries, tuples)
- implement programs using object-oriented programming
- optimize parts of their programs so as to increase their programs' efficiency

- be able to draw useful conclusions using data analysis
- have a good background to attend more advanced courses in programming.

Upon completion of the course the student should have the following learning outcomes as knowledge and skills:

- To know the principles of programming, to understand how a problem can be solved with computer programming.
- Using multiple levels of abstraction and selecting appropriate data structures to develop a program for solving a real problem and highlighting the usefulness of information technology in dealing with social problems.
- To understand the different programming styles (procedural, object-oriented, functional) and to be able to choose an appropriate programming style, to produce the desired outputs to specific inputs, to check the correctness of individual program parts and to produce documentation of the program .
- Gain experience with the event driven programming.
- Be able to use available libraries for data analysis and scientific calculations.

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 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 Project planning and management Criticism and self-criticism Production of free, creative and inductive thinking

3 COURSE CONTENT

The course aims to give students a detailed and practical introduction to basic concepts of programming and software development through a modern high-level programming language (Python). Specifically, it aims to introduce concepts such as variables, expressions, control flow, complex data structures and file processing, permanent data storage.

The laboratory consists of mandatory laboratory exercises, which are an important part of the course. In the framework of laboratory exercises, there will be guided analysis and solution of real life problems from various fields of human activity that include data management and storage, development of appropriate interaction with the user and data extraction from the internet.

4 TEACHING AND LEARNING METHODS - ASSESSMENT

<table>
<tr><td><b>TEACHING METHOD</b>
<i>Face-to-face, Distance learning, etc</i></td><td>Distance education with five Group Counseling Meetings during the academic year on weekends.</td></tr>
<tr><td><b>USE OF INFORMATION AND COMMUNICATION TECHNOLOGIES</b>
<i>Use of ICT in Teaching, in Laboratory Education, in Communication with students</i></td><td>Group Counseling Meetings and/or written assignments use:
• remote meeting tools (WebEx),
• presentation software (Powerpoint),
• specialized software (Python).

In addition, students will use office automation tools, web browsers and pdf readers for digital books.</td></tr>
<tr><td><b>TEACHING ORGANIZATION</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.</i>
<i>The student's study hours for each learning activity are given as well as the hours of nondirected study according to the principles of the ECTS</i></td><td><table>
<tr><th><i>Activity</i></th><th><i>Annual workload</i></th></tr>
<tr><td>5 Group Counseling Meetings (x 3 hours)</td><td>15</td></tr>
<tr><td>Solving 4 exercises (4 x 5 hours)</td><td>20</td></tr>
<tr><td>Project (15 weeks x 2 hours)</td><td>30</td></tr>
<tr><td>Final Examination</td><td>3</td></tr>
<tr><td>Study</td><td>92</td></tr>
<tr><td><b>Total number of hours for the Course (hours)</b></td><td><b>160</b></td></tr>
</table></td></tr>
<tr><td><b>STUDENT ASSESSEMENT</b>
<i>Description of the evaluation procedure</i>
<i>Language of evaluation, methods of evaluation, summative or conclusive, multiple choice questionnaires, short-</i></td><td>Written Assignments: Students individually prepare four written assignments, the grade of which (as an average) contributes to the final grade of the course by 30%, if there is a passable grade in the final or repeat examination.</td></tr>
</table>

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.	Examination (Final or Repeat): Students who have prepared at least three written assignments with a total score of at least 20, prepare a programming project as a group and are examined orally on it. The grade of the oral exam contributes to the final grade of the course by 70%. All the evaluation criteria are stated, both in each written assignment (on the study platform) and in the general regulation https://www.eap.gr/education/undergraduate/computer-science/topics/

5 RECOMMENDED LITERATURE

(in Greek) :

Publicly available

1) Αγγελιδάκης, Ν.Α. (2015). *Εισαγωγή στον προγραμματισμό με την Python*, [Ηλεκτρ. βιβλ.], Available at: http://aggelid.mysch.gr/pythonbook/
2) Swaroop, C. H. (2013). *A Byte of Python*, [Ηλεκτρ. βιβλ.] Επιμ. μετάφρασης Ubuntu community, Available at: http://dide.flo.sch.gr/Plinet/Meetings/Meeting23/A_Byte_of_Python-el.pdf, original: https://edu.heibai.org/A%20Byte%20of%20Python.pdf
3) **Μανής, Γ.** (2015). *Εισαγωγή στον Προγραμματισμό με αρωγό τη γλώσσα Python*. [ηλεκτρ. βιβλ.] Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Available at: http://hdl.handle.net/11419/2745
4) **Guido van Rossum and the Python Development Team.** *Python Tutorial.* Python Software Foundation. Available at: https://docs.python.org/3/tutorial/index.html

Additional digital (and multimedia) material is located on the study platform.

-Related scientific journals:

1 Science of Computer Programming, Elsevier, ISSN: 0167-6423
2 Programming and Computer Software, Springer, ISSN: 0361-7688
3 Journal of Computational Science, Elsevier, ISSN: 1877-7503
4 ACTA Informatica, Springer, ISSN: 0001-5903
5 Algorithmica, Springer, ISSN: 0178-4617