COURSE OUTLINE: COMPUTER SCIENCE II

GENERAL

SCHOOL	ECONOMICS AND BUSINESS		
ACADEMIC UNIT	ECONOMICS		
LEVEL OF STUDIES	UNDERGRADUATE		
COURSE CODE	**ΜΠ202**	**SEMESTER**	**2nd**
COURSE TITLE	**COMPUTER SCIENCE II**		

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**
	3	4
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, specialized general knowledge, skills development*	GENERAL BACKGROUND, SKILLS DEVELOPMENT
PREREQUISITE COURSES:	NO
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	GREEK
IS THE COURSE OFFERED TO ERASMUS STUDENTS	NO
COURSE WEBSITE (URL)	https://www.econ.uth.gr/

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*

Upon completion of the course, students will be able to:

- Understand the functionality of software applications, recognizing both their capabilities and limitations.
- Model systems and processes to solve problems through algorithms and programs.
- Program simple yet powerful applications using the Python programming language.
- Utilize external libraries and extensions of Python to solve complex scientific and computational 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*
	Respect for difference and multiculturalism
	Respect for the natural environment
Adapting to new situations	*Showing social, professional and ethical responsibility and sensitivity to gender issues*
Decision-making	
Working independently	*Criticism and self-criticism*
Team work	*Production of free, creative and inductive thinking*
Working in an international environment	*……*
Working in an interdisciplinary environment	*Others…*
	…….
Production of new research ideas	

- Search for, analysis and synthesis of data and information.
- Working independently
- Team work
- Project planning and management
- Production of free, creative and inductive thinking

SYLLABUS

The aim of the course is to familiarize students with algorithms and programming using the Python programming language. The course focuses on creating simple yet powerful applications that help students develop algorithmic thinking and understand how computers function. Although introductory in nature, the course is specifically tailored to meet the needs of students in economics. Python, known for its user-friendliness and high efficiency, is the primary language used in the course. This course lays the foundation for more advanced topics such as Data Analytics and Financial Technologies (FinTech), which also rely on Python.

The main course topics are:

- Introduction to programming and the basic operation of computers
- Introduction to algorithmic thinking and designing solutions through algorithms
- Familiarization with pseudocode to model algorithms
- Presentation of modern development environments (IDEs) to facilitate code development
- Learning the Python programming language and its basic syntax elements
- Using variables and operators to perform arithmetic and logical operations
- Control statements (if, else) and loop structures (for, while) to manage program flow
- Sequential data structures, such as lists, tuples, and sets, for storing and managing data
- Creating and using functions to organize and reuse code
- Introduction to Classes and Objects for object-oriented programming
- File management for storing and reading data from external sources
- Integration of Python extensions and additional libraries to enhance code functionality

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 the eclass online platform for posting (a) lecture materials, (b) announcements, and (c) exercises, case studies, and relevant articles (d) use PowerPoint presentations for lectures and Jupyter Notebooks for programming examples, (e) programming exercises using Python and R within the Anaconda development environment.
TEACHING METHODS	

Activity	***Semester workload***
Lectures	39

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*	Laboratory practice: 61 Study and analysis of bibliography: 18 Exams: 2 Course total: ***120***
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 course grade is determined by the assessment of the laboratory assignment, which accounts for 100% of the final grade. The above assessment criteria are communicated to students in the first lecture of the course and remain continuously accessible through relevant announcements on the course's eclass platform

ATTACHED BIBLIOGRAPHY

- Dimitrios Karolidis. Learn Python Easily (EUDOXUS code: 102070351).
- Harvey M. Deitel, Paul J. Deitel. Introduction to Python for Computer and Data Sciences (EUDOXUS code: 102070652).
- Johnny Wei-Bing Lin, Hannah Aizenman, Erin Manette Cartas Espinel, Kim Gunnerson, Joanne Liu. Python – Programming for Computer and Data Sciences, with scientific editing by Theodoros Katsaounis (EUDOXUS code: 122086119).
- Eric Matthes. The Python Programming Language (EUDOXUS code: 94690292).

- Fotis Karampatzakis, Dimitris Karampatzakis. Structured Programming with Applications in Python (EUDOXUS code: 112693086).
- Allen B. Downey. Think Python (EUDOXUS code: 94644736).
- Additional bibliography/articles will be suggested during the lectures