

## MODULE OUTLINE

### 1. GENERAL INFORMATION

<b>SCHOOL</b>	SCHOOL OF SCIENCE AND TECHNOLOGY		
<b>PROGRAM COURSE</b>	INFORMATICS		
<b>LEVEL OF STUDY</b>	UNDERGRADUATE		
<b>MODULE CODE</b>	PLI-11	<b>YEAR OF STUDY</b>	1 <sup>st</sup>
<b>MODULE TITLE</b>	Software engineering		
<b>INDEPENDENT TEACHING ACTIVITIES</b> <i>in case credits are awarded for separate components/parts of the course, e.g. in lectures, laboratory exercises, etc. If credits are awarded for the entire course, give the weekly teaching hours and the total credits</i>		<b>HOURS</b>	<b>CREDIS</b>
Weekly teaching hours * 32 weeks		15-16	18 ECTS
<b>COURSE TYPE</b> <i>Background knowledge, Scientific expertise, General Knowledge, Skills Development</i>	Background knowledge, Scientific expertise Compulsory		
<b>PREREQUISITE MODULES:</b>	The choose of PLI11 requires the simultaneous choose or completion of PLI10.		
<b>LANGUAGE OF INSTRUCTION AND EXAMS</b>	GREEK		
<b>THE MODULE IS OFFERED TO ERASMUS STUDENTS</b>	No (due to annual duration of the module)		
<b>MODULE WEBSITE (URL)</b>	<a href="https://www.eap.gr/education/undergraduate/computer-science/topics/#arxes_logismikou">https://www.eap.gr/education/undergraduate/computer-science/topics/#arxes_logismikou</a>  Each module has its own space in the Learning Management System of EAP ( <a href="http://study.eap.gr">http://study.eap.gr</a> ), with controlled access (use of code) for students and teaching staff.		

### 2. LEARNING OUTCOMES

<p><b>Learning Outcomes</b></p> <ul style="list-style-type: none"> <li>The course learning outcomes, specific knowledge, skills and competences of an appropriate (certain) level, which students will acquire upon successful completion of the course, are described in detail. It is necessary to consult:</li> </ul>
<p><b>SOFTWARE ENGINEERING I</b></p> <p>On successful completion of the study of this volume, students will be able to:</p> <ul style="list-style-type: none"> <li>Describe and explain the basic concepts of software engineering (software, tools, procedures, methodologies, life cycle etc.)</li> </ul>

- Recognize the fundamental role of the basic principles of software engineering in the development of qualitative software and applications
- Describe the different phases in the development of a software application as well as the basic features of the most significant software life-cycle models.
- Identify the requirements of a software application and use the principles of structured analysis for their detailed specification.
- Use effectively the most known diagrammatic software representation models (data-flow diagrams, state transition diagrams etc.) for the requirements analysis of a software application.
- Design a software application following the principles of structured design (architectural design, interface design, detailed design of modules etc.).
- Explain and apply suitable fault avoidance techniques during the implementation of a software application in order to produce qualitative code without bugs.
- Describe the different stages and explain the different strategies followed during the testing process of a software module or system.

#### *DATABASES*

On successful completion of the study of this volume, students will be able to:

- Describe the basic concepts and models of databases as well as the main differences between a database and a database management system.
- Recognize the fundamental need of using database technology in any application that demands efficient organization and management of large-scale data.
- Explain the different methods for data organization and data access (records, indexes, b-trees etc.) in modern database systems.
- Distinguish between the different levels of database analysis and design (conceptual, logical and physical level).
- Use effectively the entity-relationship model for the conceptual design of a database.
- Perform the design of a database in the logical level with use of the relational model.
- Use effectively a theoretical query language (i.e. relational algebra) for data retrieval out of a relational database.
- Implement a database over a modern relational database management system and practically use a standard query language(i.e. SQL)for efficient data organization, management and retrieval.

#### *OPERATING SYSTEMS I*

On successful completion of the study of this volume, students will be able to:

- Describe the basic functions of an operating system.

- Recognize the fundamental role of operating systems in the efficiency of modern systems and applications due to their advanced multi-tasking and multi-user capabilities.
- Describe the different methods used for process scheduling in modern operating systems and explain their differences, advantages and disadvantages. .
- Explain the mutual exclusion problem and use effectively the basic mechanisms offered by a multitasking operating system for process synchronization and communication (semaphores etc.).
- Describe the different main-memory organization methods followed in modern computer systems and identify their advantages and disadvantages.
- Generalize the use of basic memory organization methods (i.e. paging, segmentation) to more complex hybrid memory schemes, which are mainly used in modern computer systems.
- Explain the function of virtual memory organization and distinguish between the different page replacement algorithms used in modern operating systems.
- Solve practical problems and exercises with regard to the above concepts and mechanisms of a modern operating system (process management, CPU scheduling, process synchronization and communication, memory management, virtual memory organization).

### **General Competences**

*Taking into consideration the general competences that students/graduates must acquire (as those are described in the Diploma Supplement and are mentioned below), at which of the following does the course attendance aim?*

<i>Search for, analysis and synthesis of data and information by the use of appropriate technologies,</i>	<i>Project planning and management</i>
<i>Adapting to new situations</i>	<i>Respect for diversity and multiculturalism</i>
<i>Decision-making</i>	<i>Environmental awareness</i>
<i>Individual/Independent work</i>	<i>Social, professional and ethical responsibility and sensitivity to gender issues</i>
<i>Group/Team work</i>	<i>Critical thinking</i>
<i>Working in an international environment</i>	<i>Development of free, creative and inductive thinking</i>
<i>Working in an interdisciplinary environment (Other.....citizenship, spiritual freedom, social</i>	<i>.....</i>
<i>Introduction of innovative research</i>	<i>awareness, altruism etc.) .....</i>

Search for, analysis and synthesis of data and information by the use of appropriate technologies,  
 Adapting to new situations  
 Decision-making  
 Individual/Independent work  
 Project planning and management  
 Critical thinking  
 Development of free, creative and inductive thinking

### **3. MODULE CONTENT**

The main objective of the module is to introduce students to Informatics as an applied engineering science, that provides a set of documented principles, methodologies and techniques, with the help of which one can develop quality software products. Students will

first be trained in the basic principles and concepts of Software Engineering, and then they will study specific issues and application areas of the methodologies it includes. Two very important and widespread application areas of Informatics are Operating Systems and Databases. Furthermore, students will study techniques for assessing the complexity and practical application of computer-based problem-solving algorithms. In addition to the transfer of knowledge and the acquisition of skills for the application of Software Engineering methodologies, the module aims at the adoption by students of a positive attitude towards the application of widespread principles and methodologies in software development. The ultimate goal is to improve the quality of software, develop a spirit of collaboration and meet the needs for which the software system was developed.

The key subjects of the module are:

- A. Software Engineering I
- B. Operating Systems I
- C. Databases

#### 4. TEACHING METHODS--ASSESSMENT

<b>MODES OF DELIVERY</b> <i>Face-to-face, in-class lecturing, distance teaching and distance learning etc.</i>	Distance education with five Group Counseling Meetings (OSS) during the academic year on weekends.													
<b>USE OF INFORMATION AND COMMUNICATION TECHNOLOGY</b> <i>Use of ICT in teaching, Laboratory Education, Communication with students</i>	<p>We use :</p> <p>Remote meetings tools (cisco webex), Presentation software (e.g. power point), Specialized software in the subjects under study (SQLite, Diagram Drawing Tools, etc.).</p> <p>Additionally, the students use office automation tools, web browsers and e-reader for digital books.</p>													
<b>MODULE DESIGN</b> <i>Description of teaching techniques, practices and methods: Lectures, seminars, laboratory practice, fieldwork, study and analysis of bibliography, tutorials, Internship, Art Workshop, Interactive teaching, Educational visits, projects, Essay writing, Artistic creativity, etc</i>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #e0e0e0; text-align: center;"><i>Activity</i></th> <th style="background-color: #e0e0e0; text-align: center;"><i>Annual Workload</i></th> </tr> </thead> <tbody> <tr> <td>5 OSS (* 4 hours)</td> <td style="text-align: center;">20</td> </tr> <tr> <td>32 Horizontal tutorial OSS (* 2 hours)</td> <td style="text-align: center;">64</td> </tr> <tr> <td>Multiple Choice Exercises (3 x 4 hours)</td> <td style="text-align: center;">12</td> </tr> <tr> <td>Preparation of Assignments (4 assignments * 10 hours)</td> <td style="text-align: center;">40</td> </tr> <tr> <td>Examination</td> <td style="text-align: center;">3</td> </tr> </tbody> </table>		<i>Activity</i>	<i>Annual Workload</i>	5 OSS (* 4 hours)	20	32 Horizontal tutorial OSS (* 2 hours)	64	Multiple Choice Exercises (3 x 4 hours)	12	Preparation of Assignments (4 assignments * 10 hours)	40	Examination	3
<i>Activity</i>	<i>Annual Workload</i>													
5 OSS (* 4 hours)	20													
32 Horizontal tutorial OSS (* 2 hours)	64													
Multiple Choice Exercises (3 x 4 hours)	12													
Preparation of Assignments (4 assignments * 10 hours)	40													
Examination	3													

<p><i>The study hours for each learning activity as well as the hours of selfdirected study are given following the principles of the ECTS.</i></p>	Individual study	341-373
	<b>Total module workload (hours)</b>	<b>480-512</b>
<p><b>STUDENT PERFORMANCE EVALUATION/ASSESSMENT METHODS</b></p> <p><i>Detailed description of the evaluation procedures.</i></p> <p><i>Language of evaluation, assessment methods, formative or summative (conclusive), multiple choice tests, short- answer questions, open-ended questions, problem solving, written work, essay/report, oral exam, presentation, laboratory work, other.....etc.</i></p> <p><i>Specifically defined evaluation criteria are stated, as well as if and where they are accessible by the students</i></p>	<p>Elaboration of written assignments during the academic year, the average of the grades of which participates in the formation of the final grade of module by 30%, if there is a passable in the final or repetitive examinations. In the final written exams the grade of the written assignments participates in the formation of the final grade of module by 70%.</p> <p>All the criteria are posted, both in each written assignment (in the LMS study.eap.gr), as well as in the general regulation of HOU at:  <a href="https://www.eap.gr/education/study-regulations/">https://www.eap.gr/education/study-regulations/</a></p>	

## (6) SUGGESTED BIBLIOGRAPHY

<p><i>- Suggested bibliography:</i></p> <p>HOU Publications:</p> <p>A. Software Engineering I, HOU, Patras, 2000. PLI11/1</p> <p>B. Operating Systems I, HOU, Patras, 2001. PLI11/2</p> <p>C. Databases, HOU, Patras, 2000. PLI11/3</p> <p>Supplementary Document: Ch. 9 "Object-oriented Approach" of Vol. PLI10/2</p> <p>Supplementary Book: V. Verykios, "Database Lessons", Tziolas pub., Athens, 2015</p> <p>Additional digital (and multimedia) material is available within the "study" platform</p> <p><i>-Related scientific Journals:</i></p> <ol style="list-style-type: none"> <li>1) IEEE Transactions on Software Engineering</li> <li>2) Empirical Software Engineering, Springer</li> <li>3) ACM Transactions on Software Engineering &amp; Methodology</li> <li>4) Journal of Systems and Software, Elsevier</li> <li>5) ACM Transactions on Database Systems</li> </ol>
---

- 6) The VLDB Journal, Springer
- 7) ACM SIGMOD Record
- 8) ACM SIGOPS Operating Systems Review