COURSE SYLLABUS

<table>
<tr><td><b>INSTITUTION</b></td><td colspan="6">NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS</td></tr>
<tr><td><b>SCHOOL</b></td><td colspan="6">SCHOOL OF SCIENCE</td></tr>
<tr><td><b>DEPARTMENT</b></td><td colspan="6">INFORMATICS AND TELECOMMUNICATIONS</td></tr>
<tr><td><b>COURSE LEVEL</b></td><td colspan="6">UNDERGRADUATE</td></tr>
<tr><td><b>COURSE TITLE</b></td><td colspan="6"><b>Principles of Programming Languages</b></td></tr>
<tr><td><b>COURSE CODE</b></td><td colspan="2"><b>ΘΠ01</b></td><td><b>Semester</b></td><td><b>5</b></td><td><b>ECTS</b></td><td><b>6</b></td></tr>
<tr><td><b>TEACHING HOURS per week</b></td><td><b>THEORY</b></td><td><b>3</b></td><td><b>SEMINAR.</b></td><td><b>1</b></td><td><b>LABORATORY</b></td><td></td></tr>
<tr><td><b>COURSE TYPE</b></td><td colspan="6"><b>Select one of the following and delete the rest</b>
Track Compulsory (EYM)
<table>
<tr><th>K</th><th>E1</th><th>E2</th><th>E3</th><th>E4</th><th>E5</th><th>E6</th></tr>
<tr><td>A</td><td>B</td><td>B</td><td>E</td><td></td><td></td><td></td></tr>
</table>
<i>Fill the table as in the curriculum:</i> Track (A-Computer Science, B- Computer Engineering) / Specialization Compulsory (Y) / Core Specialization (B)/ Elective Specialization (E)</td></tr>
<tr><td><b>URL</b></td><td colspan="6">https://di.uoa.gr/~prondo/languages.html</td></tr>
<tr><td><b>EXPECTED PRIOR KNOWLEDGE/ PREREQUISITES AND PREPARATION:</b></td><td colspan="6">K08</td></tr>
<tr><td><b>TEACHING AND EXAMINATIONS LANGUAGE:</b></td><td colspan="6">GREEK</td></tr>
<tr><td><b>THE COURSE IS OFFERED TO ERASMUS STUDENTS</b></td><td colspan="6">NO</td></tr>
</table>

COURSE CONTENT

The course covers basic and advanced techniques of the principles of programming languages that are needed in various branches of Computer Science. History of programming languages. Syntax and Semantics. Data types, scope, calling conventions, implementation of functions and procedures. Basic categories of programming languages: imperative, object-oriented, logic, functional, parallel. Program transformations and optimizations. Proving program correctness using Hoare logic. Lambda calculus: reductions, normal forms, Church-Rosser theorem, expressiveness, applications in programming languages. Typed lambda calculus. Theory of types and type checking.

STUDENT LEARNING OBJECTIVES

Teaching-Learning Goals-Expected Learning Outcomes

The goal of the class is for the student to acquire knowledge in the Theory of Computation that is necessary in computer science. Upon successful completion of the course the student will be able to:

- Distinguish the different categories of programming languages.
- Accurately define the syntax of a programming language.
- Accurately define the semantics of simple constructs of a programming language.
- Prove that a program is correct.
- Implement problems in functional and logic programming.

<table>
<tr><th colspan="2">TEACHING AND LEARNING METHODS - ASSESSMENT</th></tr>
<tr><td><b>TEACHING METHOD</b></td><td>In Class (Face to Face)</td></tr>
<tr><td><b>USE OF INFORMATION AND COMMUNICATION TECHNOLOGIES</b></td><td>Learning process supported by the web-page of the course (Teaching material; Announcements; Task assignments; Outside links, etc)
Email communication. There exists a possibility of lecture transmission.</td></tr>
<tr><td><b>TEACHING ORGANIZATION</b>
<i>Describe in detail the way and methods of teaching:</i>
<i>Enhanced Lectures,</i>
<i>Online Lectures,</i>
<i>Seminars,</i>
<i>Tutorial,</i>
<i>Laboratory,</i>
<i>Laboratory Exercise,</i>
<i>Study &amp; analysis of literature,</i>
<i>Practice (Positioning),</i>
<i>Interactive teaching,</i>
<i>Developing a project,</i>
<i>Individual / group work</i>
<i>Telework (reference to tools) etc.</i>

<i>Details of the student's study hours for each learning activity and hours of non-guided study are shown to ensure that the total workload at the semester corresponds to the ECTS</i></td><td>
<table>
<tr><th>Activity</th><th><i>Student Workload</i>
<i>(hours)</i></th></tr>
<tr><td>Lectures</td><td>52</td></tr>
<tr><td>Seminars</td><td>13</td></tr>
<tr><td>Preparation for seminars</td><td>15</td></tr>
<tr><td>Study and analysis of the literature</td><td>15</td></tr>
<tr><td>Preparation for the next lecture</td><td>13</td></tr>
<tr><td>Homework assignments</td><td>12</td></tr>
<tr><td>Independent Study</td><td>30</td></tr>
<tr><td><b><i>Total Course</i></b>
<b><i>(25 hours of workload per unit of credit)</i></b></td><td><b><i>150</i></b></td></tr>
</table>

Lectures are supported by transparencies. The board is also used extensively. An emphasis is placed both during the lectures and the seminars on problem solving. Homework assignments are individual or in groups of 2.</td></tr>
</table>

<table>
<tr>
<td><b>ASSESSMENT OF STUDENTS</b>
<i>Description of the assessment process</i>
<i>Assessment Methods, Formative or Concluding, Multiple Choice Test, Quick Response Questions, Test Development Questions, Problem Solving, Written Work, Report / Report, Oral Examination, Public Presentation, Laboratory Work, Other / Other</i>
<i>Fully defined evaluation criteria are mentioned and if and where they are accessible to students.</i></td>
<td>
<table>
<tr><th>Assessment methods</th><th><i>Number</i></th><th><i>Percentage</i></th></tr>
<tr><td>Written examination</td><td>1</td><td>70%</td></tr>
<tr><td>Homeworks</td><td>2</td><td>30%</td></tr>
</table>
Evaluation by written examination and homework assignments.
Grade Feedback is available upon request.
</td>
</tr>
</table>

LITERATURE AND STUDY MATERIALS / READING LIST

Basic textbooks in Greek: M. Scott, Pragmatics of Programming Languages, Kleidarithmos editions, Athens 2008.

- Adam Brooks Webber, Modern Programming Languages: A Practical Introduction, Crete University Press, 2009.
- P. Stamatopoulos, Logic and Functional Programming, 2016.
- N. Papaspyrou and P. Rondogiannis, Principles of Programming Languages (notes).

Additionally, the students have access to transparencies by P. Rondogiannis, and recommended literature in English.