Educational programming environments

SCHOOL	School of Humanities		
ACADEMIC UNIT	Department of Primary Education		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	EG0045	**SEMESTER**	Spring
COURSE TITLE	Educational programming environments		
INDEPENDENT TEACHING ACTIVITIES		**WEEKLY TEACHING HOURS**	**CREDITS**
		3	4
COURSE TYPE	special background, skills development, lab, elective		
COURSE TYPE:	None		
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	Greek		
IS THE COURSE OFFERED TO ERASMUS STUDENTS	Yes		
COURSE WEBSITE (URL)	-		

LEARNING OUTCOMES

By the end of this course, students will be able.

In terms of knowledge:

- To become familiar with basic programming concepts.
- To understand how programs function.
- To know the steps required to develop a program.
- To develop complex programs.

In terms of their skills:

- To be skillful in creating and solving algorithms.
- To encode/transform algorithms in commands and programs.
- To be able to choose the right programming environment for materializing their objectives.

In terms of their competences:

- To be able to develop educational applications using programming environments
- To be able to modify existing educational programs and tailor them to the needs of their students.
- To form a positive attitude towards programming for educational use

General Competences

The course aims at the following general competences:

- Adapting to new situations
- Decision-making
- Working independently
- Team work
- Working in an interdisciplinary environment
- Production of new research ideas

Learning how to program has some noteworthy advantages for students. That is because it helps them to acquire analytic and synthetic thinking skills in order to be able to solve problems in a programming environment. It also helps them to understand how algorithms are developed and used (problem analysis, algorithm's design, structured thinking, rigorous expression) and it has a positive impact on creativity and imagination. Learning how to program, as a teaching/learning subject, is based on the Interdisciplinary Informatics Curriculum Framework and it is taught in the last two grades in primary school.

Έτσι, βασικός στόχος του μαθήματος είναι η αξιοποίηση του προγραμματισμού ως γνωστικού εργαλείου για το μάθημα της Πληροφορικής, αλλά και η διερεύνηση τρόπων ένταξής του σε άλλα μαθήματα του δημοτικού σχολείου. Πρέπει να τονιστεί ότι το μάθημα στηρίζεται σε προγραμματιστικά περιβάλλοντα κατάλληλα για την πρωτοβάθμια εκπαίδευση, πράγμα που σημαίνει ότι μπορούν να τα χρησιμοποιήσουν τόσο οι εκπαιδευτικοί όσο και οι μαθητές. Επίσης, έμφαση δίνεται σε ελεύθερα περιβάλλοντα οπτικού προγραμματισμού με πλακίδια, όπως το Scratch, και το Scratch JR. Το μάθημα περιλαμβάνει θεωρητικό και εργαστηριακό μέρος και περιλαμβάνει τους παρακάτω άξονες/θεματικές ενότητες:

On the basis of the above, the aim of the course is to provide students the necessary knowledge and skills on the basic programming principles and techniques, in the context of computer literacy. Thus, the main objective is for pre-service teachers to understand how programming can be utilized in the context of primary school's ICT courses, but also to explore how programming can be integrated to other courses. It should be noted that the course is based on programming environments suitable for primary education, for both teachers and students. The course includes a theoretical and a hands-on part and includes the following modules

- Basic programming concepts.
- Programming as a teaching subject in primary school.
- The programming environment of Scratch and Scratch JR.

- Implementation methodology of basic computer programming concepts in Scratch.
- Application development methodology in Scratch.
- Development of educational software in Scratch.
- Design and implementation of teaching scenarios using Scratch.

<table>
<tr><td><b>DELIVERY</b></td><td>Face-to-face using PCs and/or laptops</td></tr>
<tr><td><b>SE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY</b></td><td>Yes
ICT is the subject of the course</td></tr>
<tr><td><b>TEACHING METHODS</b></td><td>
<table>
<tr><th><i>Activity</i></th><th><i>Semester workload</i></th></tr>
<tr><td>Lectures</td><td>25</td></tr>
<tr><td>Lab exercises</td><td>25</td></tr>
<tr><td>Independent study</td><td>30</td></tr>
<tr><td>Application development</td><td>40</td></tr>
<tr><td><b>Course total</b></td><td><b><i>120</i></b></td></tr>
</table>
</td></tr>
<tr><td><b>STUDENT PERFORMANCE EVALUATION</b></td><td>The evaluation encompasses the following components:
1. Laboratory exercises conducted throughout the semester.
2. Written midterm examination addressing theoretical issues related to the teaching of programming in students.
3. Each student is tasked with designing and implementing an application, either individually or in a small group, utilizing the tools discussed in the lectures. The assessment of this application is based on specific criteria, including the pedagogical, technical, and aesthetic adequacy of the material produced.
The successful completion of all the aforementioned activities is a prerequisite for the awarding of academic credits.</td></tr>
</table>

- Suggested bibliography:

Teachers’ documents, tutorials and scientific articles published to Moodle

- Related academic journals:

- Computers and Education
- International Journal of Game-Based Learning
- Education and Information Technologies

- Australasian Journal of Educational Technology
- Journal of Educational Technology & Society