

340368 - FOPR-I1023 - Fundamentals of Programming

Coordinating unit:	340 - EPSEVG - Vilanova i la Geltrú School of Engineering
Teaching unit:	723 - CS - Department of Computer Science
Academic year:	2019
Degree:	BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2018). (Teaching unit Compulsory) BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Teaching unit Compulsory)
ECTS credits:	7,5
Teaching languages:	Catalan

Teaching staff

Coordinator:	ÀNGELS HERNÁNDEZ GÓMEZ
Others:	ÀNGELS HERNÁNDEZ GÓMEZ BERNARDINO CASAS FERNÁNDEZ

Prior skills

Basic knowledge of mathematics for the required level at the university entrance exam.

Degree competences to which the subject contributes

Specific:

3. CEFB3. Ability to understand and to have a good command of discrete, logical, algorithmically mathematics and computing complexity and its application to automatical treatment of information by means of computational systems and its application to solve engineering problems.
4. CEFB4. Basic knowledge of use and computer programming, as well as of operating systems, data base and generally informatic programs with engineering applications.
5. CEFC6. Basic knowledge and application of algorithmic processes, informatic techniques to design solutions of problems, analyzing if proposed algorithms are apt and complex.
6. CEFC7. Knowledge, design and efficient use of data types and structures the most appropriate to resolve problems.

Transversal:

1. SELF-DIRECTED LEARNING. Detecting gaps in one's knowledge and overcoming them through critical self-appraisal. Choosing the best path for broadening one's knowledge.
2. TEAMWORK. Being able to work as a team player, either as a member or as a leader. Contributing to projects pragmatically and responsibly, by reaching commitments in accordance to the resources that are available.

Teaching methodology

The course consists of:

- 2 hours per week of theory class in the classroom (large group) where the teacher presents contents,
- 3 hours per week in classroom lab (small group) where it's done practical work and evaluable activities are proposed and performed.

Learning objectives of the subject

1. To understand the programme building process and how to use the tools required: console, editor and compiler.
2. To know the syntax and the semantics expressions and the basic instructions of the imperative programming languages (C ++).
3. Being proficient in using functions and actions in programming.
4. To understand the function and parameter concepts.

340368 - FOPR-I1023 - Fundamentals of Programming

5. To know in depth the tables and to identify where their use is appropriate.
6. To be able to contrast solutions regarding the use of time and memory resources and to choose the most appropriate in simple cases.
7. To understand the patterns of treat-all and search algorithms patterns.
8. Choosing an appropriate scheme solution.
9. To understand the recursion concept. To be able to propose recursive solutions to simple problems.
10. To comprehend in depth the binary search, insertion sort, selection sort, mergesort and quicksort.
11. To know in depth other fundamental algorithms: Hörner, fast product, etc.
12. To be able to write programs readable, elegant and efficient.

Study load

Total learning time: 187h 30m	Hours large group:	30h	16.00%
	Hours medium group:	0h	0.00%
	Hours small group:	45h	24.00%
	Guided activities:	0h	0.00%
	Self study:	112h 30m	60.00%

340368 - FOPR-I1023 - Fundamentals of Programming

Content

<p>Introduction</p>	<p>Learning time: 6h Theory classes: 2h Laboratory classes: 3h Self study : 1h</p>
<p>Description: Programming examples Algorithms, programming languages and computer programs Steps in the design of a program</p>	
<p>Variables and statements</p>	<p>Learning time: 6h Theory classes: 2h Laboratory classes: 3h Self study : 1h</p>
<p>Description: Variables, data types and expressions Statements: <ul style="list-style-type: none"> - Assignment - Input / Output - Conditional statement </p>	
<p>Loops</p>	<p>Learning time: 6h Theory classes: 2h Laboratory classes: 3h Self study : 1h</p>
<p>Description: While statement For statement</p>	
<p>Data types and visibility</p>	<p>Learning time: 6h Theory classes: 2h Laboratory classes: 3h Self study : 1h</p>
<p>Description: Data types Type conversion Visibility</p>	

340368 - FOPR-I1023 - Fundamentals of Programming

Subprograms: procedures and functions	Learning time: 7h Theory classes: 2h Laboratory classes: 3h Self study : 2h
Description: Subprogram concept Parameter passing Functions Procedures	
Algorithms on sequences. Invariants.	Learning time: 7h Theory classes: 2h Laboratory classes: 3h Self study : 2h
Description: Algorithms on sequences: - Treat-all algorithms - Search algorithms Reasoning about loops: invariants	
Recursion	Learning time: 6h Theory classes: 2h Laboratory classes: 3h Self study : 1h
Description: Recursive design Exemples	
Vectors	Learning time: 6h Theory classes: 2h Laboratory classes: 3h Self study : 1h
Description: Vectors Searching in vectors	

340368 - FOPR-I1023 - Fundamentals of Programming

<p>Vectors and strings</p>	<p>Learning time: 6h Theory classes: 2h Practical classes: 3h Self study : 1h</p>
<p>Description: More vectors examples Strings</p>	
<p>Multi-dimensional vectors</p>	<p>Learning time: 8h Theory classes: 2h Laboratory classes: 4h Self study : 2h</p>
<p>Description: Matrices N-dimensional vectors Search in a matrix Search in a sorted matrix Matrix multiplication</p>	
<p>Structures and data structure design</p>	<p>Learning time: 6h Theory classes: 2h Laboratory classes: 3h Self study : 1h</p>
<p>Description: Structures Data structure design</p>	
<p>Sorting</p>	<p>Learning time: 8h Theory classes: 2h Laboratory classes: 4h Self study : 2h</p>
<p>Description: Selection Sort Insertion Sort Bubble Sort Merge Sort</p>	

340368 - FOPR-I1023 - Fundamentals of Programming

<p>Numerical algorithms</p>	<p>Learning time: 9h Theory classes: 2h Laboratory classes: 4h Self study : 3h</p>
<p>Description: Product of polynomials Sum of polynomials Sum of sparse vectors Root of a continuous function</p>	
<p>Advanced examples</p>	<p>Learning time: 7h Theory classes: 2h Laboratory classes: 3h Self study : 2h</p>
<p>Description: Sports tournament Permutations Sub-sequences summing n</p>	
<p>Conclusions</p>	<p>Learning time: 2h Theory classes: 2h</p>
<p>Description: Why is programming hard? Useful programs Correct programs Efficient programs Programs are mathematical objects Easy to understand, modify and extend Programming has limits Quotes</p>	

340368 - FOPR-I1023 - Fundamentals of Programming

Qualification system

QU = Grade from questionnaires, all with the same weight.

AC = Grade from activities, all with the same weight.

PR = Grade from programming task.

C1 = Grade from Exam 1.

C2 = Grade from Exam 2.

PV = Validation Test.

Nota Final = $\max(50\%C2, 20\%C1+30\%C2) + 10\%QU + 20\%AC + 20\%(PR*PV)$

The reevaluation contains the C2 test.

Regulations for carrying out activities

The activities (AC), the controls (C1 and C2) and the validation test (PV) are face and individual.

The questionnaires (QU) are self-assessed and their delivery is electronic and individual.

The programming task (PR) is done in groups. Teachers could ask about the work presented by the students and consider their answers when qualifying. The validation test shall be carried out in conjunction with Control 2.

Bibliography