Course guide
270001 - PRO1 - Programming I

Last modified: 30/01/2024

Unit in charge: Barcelona School of Informatics
Teaching unit: 723 - CS - Department of Computer Science.

Degree: BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Compulsory subject).

Academic year: 2023 **ECTS Credits:** 7.5 **Languages:** Catalan, Spanish, English

LECTURER

Coordinating lecturer: GUILLERMO GODOY BALIL - PABLO FERNANDEZ DURAN

Others:

Primer quadrimestre:
JORGE CASTRO RABAL - 41, 42, 43, 71, 72, 73
PABLO FERNANDEZ DURAN - 11
GUILLERMO GODOY BALIL - 11, 12, 13
ALEXANDRE GRACIA CALVO - 43
ROSA MARIA JIMÉNEZ GÓMEZ - 31, 51, 71
MONTSERRAT MADRIDEJOS MORA - 53
ALEXIS MOLINA MARTINEZ DE LOS REYES - 13, 23
GLYN VERDEN MORRILL - 33, 63, 73
NICOLAS EDUARDO MYLONAKIS PASCUAL - 21, 22, 23, 32, 42
MARIA ANGELA NEBOT CASTELLS - 22, 31, 32, 33, 52, 62
LLUIS PADRO CIRERA - 12, 51, 52, 53
EMMA ROLLÓN RICO - 41, 61, 62, 63
XAVIER RUBIÉS CULLELL - 72
ALFONSO VALVERDE RUIZ - 21, 61

Segon quadrimestre:
ROSA MARIA JIMÉNEZ GÓMEZ - 11, 21, 22
GLYN VERDEN MORRILL - 12, 21
EMMA ROLLÓN RICO - 11, 12

PRIOR SKILLS

Secondary school.

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Specific:
CT1.1A. To demonstrate knowledge and comprehension about the fundamentals of computer usage and programming, about operating systems, databases and, in general, about computer programs applicable to the engineering.
CT1.1B. To demonstrate knowledge and comprehension about the fundamentals of computer usage and programming. Knowledge about the structure, operation and interconnection of computer systems, and about the fundamentals of its programming.
CT1.2B. To interpret, select and value concepts, theories, uses and technological developments related to computer science and its application derived from the needed fundamentals of mathematics, statistics and physics. Capacity to understand and dominate the physical and technological fundamentals of computer science: electromagnetism, waves, circuit theory, electronics and photonics and its application to solve engineering problems.
CT4.1. To identify the most adequate algorithmic solutions to solve medium difficulty problems.
CT4.2. To reason about the correction and efficiency of an algorithmic solution.
CT5.2. To know, design and use efficiently the most adequate data types and data structures to solve a problem.
CT5.3. To design, write, test, refine, document and maintain code in an high level programming language to solve programming problems applying algorithmic schemas and using data structures.
CT5.4. To design the programs¿ architecture using techniques of object orientation, modularization and specification and implementation of abstract data types.
CT8.6. To demonstrate the comprehension of the importance of the negotiation, effective working habits, leadership and communication skills in all the software development environments.

Generical:
G7. AUTONOMOUS LEARNING: to detect deficiencies in the own knowledge and overcome them through critical reflection and choosing the best actuation to extend this knowledge. Capacity for learning new methods and technologies, and versatility to adapt oneself to new situations.

TEACHING METHODOLOGY

In the theory sessions, the lecturer will alternate new theoretical concepts with examples and exercises. Lectures, in which the course topics are presented, explained and illustrated, will be combined with student interaction regarding the various alternatives arising in the resolution of practical cases.

The laboratory sessions have two distinct parts:
During the first hour, a guided session takes place, where the lecturer describes practical issues regarding the programming environment, or some exercises are solven in a collaborative way, or some code is analyzed to identify errors, etc.
Then students devote the remaining two hours to solve problems with the automatic judge with the assistance of the lecturer if needed.

Students are expected, in the laboratory sessions and in home study, to resolve problems from a set of problems and upload their solutions to an automatic judge for checking by comprehensive test suites. They are also advised to regularly consult their lecturer about their programs (irrespective of whether they work) for an evaluation of their quality.

LEARNING OBJECTIVES OF THE SUBJECT

1.Understand how to build a program and use tools the necessary tools: console, editor and compiler.
2.Understand the syntax and semantics of basic expressions and instructions in an imperative programming language (C++).
3.Use functions and actions to develop programs.
4.Understand the concepts of function, action and parameter passing
5.Understand tables and identify problems for which their use is appropriate.
6.Compare solutions regarding time and memory use and choose the most appropriate solutions for simple cases.
7.Understand search and traversal diagrams.
8.Associate a problem with an appropriate solution scheme
9.Understand recursion and develop recursive solutions for simple problems.
10.Understand binary search, insertion, sorting, selection, mergesort and quicksort algorithms.
11.Understand other fundamental algorithms: Hörner, fast product
12.Write programs of about one page in length that are readable, efficient and elegant.

STUDY LOAD

Type	Hours	Percentage
Hours large group	30,0	16.00
Hours small group	45,0	24.00
Self study	105,0	56.00
Guided activities	7,5	4.00

Total learning time: 187.5 h

CONTENTS

Basic programming principles

Description:
Introduction to fundamental concepts: algorithm, program, variable, expression, data type, etc. Basic C++ instructions.

Iterative instructions

Description:
For and while instructions. Examples.

Traversal and search diagrams

Description:
Sequences. Sequential traversal and search.

Actions and functions

Description:
Actions and functions. Parameter passing. Visibility levels.

Recursion

Description:
Introduction to recursive design.

Tables

Description:
One-dimensional tables. Multidimensional tables. Traversals and searches in tables.

Tuples

Description:
Programming with tuples.

Basic algorithms I

Description:
Sorting algorithms. Binary search.

Basic algorithms II

Description:
Other important algorithms: Hörner, fast product, etc.

ACTIVITIES

Topic development: Basic programming principles

Description:
Understand and assimilate the concepts covered in theory classes. Solve the problems set for this topic, available at www.jutge.org.

Specific objectives:
1, 2

Related competencies :
G7. AUTONOMOUS LEARNING: to detect deficiencies in the own knowledge and overcome them through critical reflection and choosing the best actuation to extend this knowledge. Capacity for learning new methods and technologies, and versatility to adapt oneself to new situations.

Full-or-part-time: 11h
Theory classes: 2h
Laboratory classes: 3h
Self study: 6h

Topic development: Iterative instructions

Description:
Understand and assimilate the concepts covered in theory classes. Solve the problems set for this topic, available at www.jutge.org.

Specific objectives:
1, 2

Related competencies :
G7. AUTONOMOUS LEARNING: to detect deficiencies in the own knowledge and overcome them through critical reflection and choosing the best actuation to extend this knowledge. Capacity for learning new methods and technologies, and versatility to adapt oneself to new situations.

Full-or-part-time: 11h
Theory classes: 2h
Laboratory classes: 3h
Self study: 6h

Topic development: Traversal and search schemes

Description:
Understand and assimilate the concepts covered in theory classes. Solve the problems set for this topic, available at www.jutge.org.

Specific objectives:
2, 7, 8

Full-or-part-time: 11h
Theory classes: 2h
Laboratory classes: 3h
Self study: 6h

Topic development: Actions and functions.

Description:
Understand and assimilate the concepts covered in theory classes. Solve the problems set for this topic, available at www.jutge.org.

Specific objectives:
2, 3, 4

Related competencies :
G7. AUTONOMOUS LEARNING: to detect deficiencies in the own knowledge and overcome them through critical reflection and choosing the best actuation to extend this knowledge. Capacity for learning new methods and technologies, and versatility to adapt oneself to new situations.

Full-or-part-time: 11h
Theory classes: 2h
Laboratory classes: 3h
Self study: 6h

Topic development: "Recursion"

Description:
Understand and assimilate the concepts covered in theory classes. Solve the problems set for this topic, available at www.jutge.org.

Specific objectives:
3, 4, 9

Related competencies :
G7. AUTONOMOUS LEARNING: to detect deficiencies in the own knowledge and overcome them through critical reflection and choosing the best actuation to extend this knowledge. Capacity for learning new methods and technologies, and versatility to adapt oneself to new situations.

Full-or-part-time: 11h
Theory classes: 2h
Laboratory classes: 3h
Self study: 6h

Consolidation: topics 1 to 5

Description:
Understand and assimilate the concepts covered in theory classes. Solve the problems set for the purpose of consolidating the first part of the course at www.jutge.org.

Specific objectives:
1, 2, 3, 4, 7, 8, 9

Related competencies :
G7. AUTONOMOUS LEARNING: to detect deficiencies in the own knowledge and overcome them through critical reflection and choosing the best actuation to extend this knowledge. Capacity for learning new methods and technologies, and versatility to adapt oneself to new situations.

Full-or-part-time: 27h 30m
Theory classes: 2h
Laboratory classes: 6h
Self study: 19h 30m

Topic development: Tables

Description:
Understand and assimilate the concepts covered in theory classes. Solve the problems set for this topic, available at www.jutge.org.

Specific objectives:
5, 6, 7, 8, 12

Related competencies :
G7. AUTONOMOUS LEARNING: to detect deficiencies in the own knowledge and overcome them through critical reflection and choosing the best actuation to extend this knowledge. Capacity for learning new methods and technologies, and versatility to adapt oneself to new situations.

Full-or-part-time: 22h
Theory classes: 4h
Laboratory classes: 6h
Self study: 12h

Topic development: Tuples

Description:
Understand and assimilate the concepts covered in theory classes. Solve the problems set for this topic, available at www.jutge.org.

Specific objectives:
12

Related competencies :
G7. AUTONOMOUS LEARNING: to detect deficiencies in the own knowledge and overcome them through critical reflection and choosing the best actuation to extend this knowledge. Capacity for learning new methods and technologies, and versatility to adapt oneself to new situations.

Full-or-part-time: 11h
Theory classes: 2h
Laboratory classes: 3h
Self study: 6h

Topic development: Basic algorithms I

Description:
Understand and assimilate the concepts covered in theory classes. Solve the problems set for this topic, available at www.jutge.org.

Specific objectives:
5, 6, 10, 12

Related competencies :
G7. AUTONOMOUS LEARNING: to detect deficiencies in the own knowledge and overcome them through critical reflection and choosing the best actuation to extend this knowledge. Capacity for learning new methods and technologies, and versatility to adapt oneself to new situations.

Full-or-part-time: 11h
Theory classes: 2h
Laboratory classes: 3h
Self study: 6h

Topic development: Basic algorithms II

Description:
Understand and assimilate the concepts covered in theory classes. Solve the problems set for this topic, available at www.jutge.org.

Specific objectives:
6, 11, 12

Related competencies :
G7. AUTONOMOUS LEARNING: to detect deficiencies in the own knowledge and overcome them through critical reflection and choosing the best actuation to extend this knowledge. Capacity for learning new methods and technologies, and versatility to adapt oneself to new situations.

Full-or-part-time: 11h
Theory classes: 2h
Laboratory classes: 3h
Self study: 6h

Consolidation

Description:
Solve the problems set for this topic, available at www.jutge.org.

Specific objectives:
12

Related competencies :
G7. AUTONOMOUS LEARNING: to detect deficiencies in the own knowledge and overcome them through critical reflection and choosing the best actuation to extend this knowledge. Capacity for learning new methods and technologies, and versatility to adapt oneself to new situations.

Full-or-part-time: 42h
Laboratory classes: 9h
Self study: 33h

Midterm exam

Description:
Several programming exercises should be solved using the jutge.org platform.

Specific objectives:
1, 2, 8

Related competencies :
G7. AUTONOMOUS LEARNING: to detect deficiencies in the own knowledge and overcome them through critical reflection and choosing the best actuation to extend this knowledge. Capacity for learning new methods and technologies, and versatility to adapt oneself to new situations.

Full-or-part-time: 2h
Guided activities: 2h

Midterm exam

Description:
Several programming exercises should be solved using the jutge.org platform.

Specific objectives:
1, 2, 3, 4, 5, 7, 8, 9

Related competencies :
G7. AUTONOMOUS LEARNING: to detect deficiencies in the own knowledge and overcome them through critical reflection and choosing the best actuation to extend this knowledge. Capacity for learning new methods and technologies, and versatility to adapt oneself to new situations.

Full-or-part-time: 3h
Guided activities: 3h

Final exam

Description:
Several programming exercises should be solved using the jutge.org platform. Global course evaluation

Specific objectives:
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

Related competencies :
G7. AUTONOMOUS LEARNING: to detect deficiencies in the own knowledge and overcome them through critical reflection and choosing the best actuation to extend this knowledge. Capacity for learning new methods and technologies, and versatility to adapt oneself to new situations.

Full-or-part-time: 3h
Guided activities: 3h

GRADING SYSTEM

There will be homework on a regular basis all along the course, consisting of delivering through judge solutions to specific exercises. The overall mark arising from the homework will be worth a 15% of the subject global mark. Students can take advandage of lab sessions to ask questions about the homework exercises.

Also, two exams will take place during the course: the mid-term exam and the final exam. Let's call D, P and F the marks out of 10 obtained from the homework, the mid-term exam and the final exam, respectively. The subject global mark is obtained as follows:

N = 0.15 * D + 0.85 * max ((0,4 * P + 0,6 * F), F)

An exam will have NP grade if no submission has been made to any of its problems.
The subject global grade N will be NP if both mid-term and final exams have NP grade.

RE-EVALUATION
Re-evaluation consists of an presential intensive course of 12 hours plus an evaluation, taking place after final exams and before the start of the next semester. Re-evaluation is estimated to require about 50 hours of effort, including sessions, homework, and evaluation.

Minimum requirements to be eligible for re-evaluation:

- Being enroled in the course
- Having obtained a global grade between 3.5 and 4.9

Requirements to be re-evaluated:

- Attend all sessions of the intensive course
- Do the homework and other activities requested by course professors.

Evaluation:
The result of the intensive course evaluation will be "pass" or "fail". The final grade for the course will be:
Final score = 5 if the intensive course score is "pass"
Final score = N if the intensive course score is "fail"
(where N is the grade obtained in the regular course evaluation)

GENERAL COMPETENCE

The evaluation of the general competence "Autonomous Learning" is based on A=min(D,1.5*N) and will be:

NP if the subject grade N is NP or the amount of delivered homework is less than a 50% of the total. Otherwise:
D si 0 C si 5 B si 7 A si 9

BIBLIOGRAPHY

Basic:
- Professorat de Programació 1. Transparències de teoria de l'assignatura (Pàgina web de l'assignatura, apartat material docent).
- Professorat de Programació 1. Dotze algorismes fonamentals (Pàgina web de l'assignatura, apartat material docent).

Complementary:
- King, K.N. C programming: a modern approach. 2nd ed. W.W. Norton and company, 2008. ISBN 9780393979503.
- Oualline, S. Practical C++ programming. 2nd ed. O'Reilly, 2003. ISBN 0596004192.

RESOURCES

Hyperlink:
- http://www-cs-faculty.stanford.edu/~eroberts/books/ArtAndScienceOfJava/- https://www.jutge.org- http://www.cprogramming.com/- http://www.uow.edu.au/~nabg/ABC/ABC.html- http://pro1.cs.upc.edu/- http://minidosis.org/