Course guide
270001 - PRO1 - Programming I

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: 2022 ECTS Credits: 7.5 Languages: Catalan, Spanish, English

LECTURER

Coordinating lecturer: JORGE CASTRO RABAL - EMMA ROLLÓN RICO

Others:
Primer quadrimestre:
JORGE CASTRO RABAL - 11, 12, 13, 22, 31, 41
JOAQUIN GABARRÓ VALLÉS - 71, 72, 73
MARC GÀLLEGOS ASÍN - 11, 62
ROSA MARIA JIMÉNEZ GÓMEZ - 12, 22, 62, 73
RAÚL LÓPEZ SÁNCHEZ - 51
MONTSERRAT MADRIDEJOS MORA - 53
ALEXIS MOLINA MARTINEZ DE LOS REYES - 23, 32
GLYN VERDEN MORRILL - 33, 63
NICOLAS EDUARDO MYLONAKIS PASCUAL - 21, 42
MARIA ANGELA NEBOT CASTELLS - 13, 21, 22, 31, 32, 33, 41, 42, 43, 52
ANA EDELMIRO PASCARELLA SANCHEZ - 71
EMMA ROLLÓN RICO - 11, 12, 13, 41, 51, 52, 53, 61, 62, 63, 73
XAVIER RUBIÉS CULLELL - 72
ALFONSO VALVERDE RUIZ - 43, 61

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 program's 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 solved 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
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**

<table>
<thead>
<tr>
<th>Type</th>
<th>Hours</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guided activities</td>
<td>7,5</td>
<td>4.00</td>
</tr>
<tr>
<td>Self study</td>
<td>105,0</td>
<td>56.00</td>
</tr>
<tr>
<td>Hours large group</td>
<td>30,0</td>
<td>16.00</td>
</tr>
<tr>
<td>Hours small group</td>
<td>45,0</td>
<td>24.00</td>
</tr>
</tbody>
</table>

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.
<table>
<thead>
<tr>
<th><strong>Iterative instructions</strong></th>
<th>Description: For and while instructions. Examples.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Traversal and search diagrams</strong></td>
<td>Description: Sequences. Sequential traversal and search.</td>
</tr>
<tr>
<td><strong>Actions and functions</strong></td>
<td>Description: Actions and functions. Parameter passing. Visibility levels.</td>
</tr>
<tr>
<td><strong>Recursion</strong></td>
<td>Description: Introduction to recursive design.</td>
</tr>
<tr>
<td><strong>Tables</strong></td>
<td>Description: One-dimensional tables. Multidimensional tables. Traversals and searches in tables.</td>
</tr>
<tr>
<td><strong>Tuples</strong></td>
<td>Description: Programming with tuples.</td>
</tr>
<tr>
<td><strong>Basic algorithms I</strong></td>
<td>Description: Sorting algorithms. Binary search.</td>
</tr>
<tr>
<td><strong>Basic algorithms II</strong></td>
<td>Description: Other important algorithms: Hörner, fast product, etc.</td>
</tr>
</tbody>
</table>
## ACTIVITIES

<table>
<thead>
<tr>
<th>Topic development: Basic programming principles</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong> Understand and assimilate the concepts covered in theory classes. Solve the problems set for this topic, available at <a href="http://www.jutge.org">www.jutge.org</a>.</td>
</tr>
<tr>
<td><strong>Specific objectives:</strong> 1, 2</td>
</tr>
<tr>
<td><strong>Related competencies:</strong> 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.</td>
</tr>
</tbody>
</table>
| **Full-or-part-time:** 11h  
Theory classes: 2h  
Laboratory classes: 3h  
Self study: 6h |

<table>
<thead>
<tr>
<th>Topic development: Iterative instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong> Understand and assimilate the concepts covered in theory classes. Solve the problems set for this topic, available at <a href="http://www.jutge.org">www.jutge.org</a>.</td>
</tr>
<tr>
<td><strong>Specific objectives:</strong> 1, 2</td>
</tr>
<tr>
<td><strong>Related competencies:</strong> 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.</td>
</tr>
</tbody>
</table>
| **Full-or-part-time:** 11h  
Theory classes: 2h  
Laboratory classes: 3h  
Self study: 6h |

<table>
<thead>
<tr>
<th>Topic development: Traversal and search schemes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong> Understand and assimilate the concepts covered in theory classes. Solve the problems set for this topic, available at <a href="http://www.jutge.org">www.jutge.org</a>.</td>
</tr>
<tr>
<td><strong>Specific objectives:</strong> 2, 7, 8</td>
</tr>
</tbody>
</table>
| **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 48m
- Theory classes: 2h
- Laboratory classes: 3h
- Guided activities: 0h 48m
- 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:** 22h
- Theory classes: 4h
- Laboratory classes: 6h
- Self study: 12h
**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 54m
Theory classes: 4h
Laboratory classes: 6h
Guided activities: 0h 54m
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:** 53h 48m
Theory classes: 6h
Laboratory classes: 9h
Guided activities: 5h 48m
Self study: 33h

---

**Test 1**

**Description:**
Several programming exercises should be solved using the jutge.org platform. Before being allowed to sit this test, students may be asked to individually resolve a number of exercises from a list.

**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.
Test 2

**Description:**
Several programming exercises should be solved using the judge.org platform. Before being allowed to sit this test, students may be asked to individually resolve a number of exercises from a list.

**Specific objectives:**
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.

Final exam

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

**Specific objectives:**
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.

**GRADING SYSTEM**

In this course course three exams P1, P2 and F are taken. The first two are partial exams while the last one is a final exam. Each exam has a list of associated exercises. To be allowed to take P1 and P2 it is mandatory to solve a minimum of exercises in their lists. Furthermore, solving a minimum number of exercises in the list associated with the final exam F enables the FF value of the evaluation formula below to be the F grade (otherwise, the FF value will be non-presented (NP)). The minimum of problems in each list, the delivery procedure and deadlines are shown on the course web page. The course final grade is obtained from the formula:

\[
\max (0.2 \times P1 + 0.3 \times P2 + 0.5 \times FF, F)
\]

An exam will have NP grade if no submission has been made to any of its problems.

**RE-EVALUATION**

Re-evaluation consists of an presentential 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 a value different from NP in P1, P2 and FF at the final grade formula above. Note that this requirement implies having solved a number of problems at each associated course exam lists.
- Having obtained a grade NF 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 = NF if the intensive course score is "fail"
(where NF is the grade obtained in the regular course evaluation)

GENERAL COMPETENCE

The evaluation of the general competence "Autonomous Learning" is based on 2 collected data related to student performance regarding the lists of problems to deliver during the quarter:

Self-learning effort (E):
Average ratio submitted_problem / minimum_required_problems for each exam in the course. It shall be calculated as: E = (E1 + E2 + E3) / 3. The metric is saturated in 2. This measure aims to encourage students to do more exercises than the minimum required for each exam.

Learning planning (P):
Measure distribution in time of deliveries of the problems in required lists. It is calculated as P = (P1 + P2 + ... + Pn) / n
Where:
Pi = -Xi*log(Xi) - (1-Xi)*log(1-Xi), if Xi<0.5
Pi = 1, if Xi>=0.5
where Xi is the percentage of problems presented in the first half of the delivery period of the list (and therefore 1-Xi is the percentage of problems presented in the second half of the period). A low P value indicates tendency to delay effort until the second half of the expected period. A high value indicates a more uniform distribution of the effort, or concentration of the effort in the first half of the period.

This metric aims to encourage students to organize work and distribute exercises in time, following the pace of theory and laboratory sessions.

The grade for the general competence will be:
NP if E <= 0.5 (Not enough exercises presented to evaluate competence)
D if 0.5 <E <1 (Minimum required not reached, indicating little self-learning effort)
C if E>=1 and P * E <= 0.4
B if E>=1 and 0.4 <P * E <= 1
A if E>=1 and P * E> 1

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:

RESOURCES

Hyperlink:
- http://www.cprogramming.com/
- http://www.cs-faculty.stanford.edu/~eroberts/books/ArtAndScienceOfJava/
- http://www.isi.upc.edu/~pro1/
- http://minidosis.org/
- https://www.jutge.org