DEGREE CURRICULUM

INTRODUCTION TO PROGRAMMING II

Coordination: GIMENO ILLA, JUAN MANUEL

Academic year 2022-23

INTRODUCTION TO PROGRAMMING II 2022-23

Subject's general information

<table>
<tr><td>Subject name</td><td>INTRODUCTION TO PROGRAMMING II</td></tr>
<tr><td>Code</td><td>102001</td></tr>
<tr><td>Semester</td><td>2nd Q(SEMESTER) CONTINUED EVALUATION</td></tr>
<tr><td>Typology</td><td>
<table>
<tr><th>Degree</th><th>Course</th><th>Character</th><th>Modality</th></tr>
<tr><td>Bachelor's Degree in Computer Engineering</td><td>1</td><td>COMMON/CORE</td><td>Attendance-based</td></tr>
<tr><td>Double bachelor's degree: Degree in Computer Engineering and Degree in Business Administration and Management</td><td>1</td><td>COMMON/CORE</td><td>Attendance-based</td></tr>
</table>
</td></tr>
<tr><td>Course number of credits (ECTS)</td><td>6</td></tr>
<tr><td>Type of activity, credits, and groups</td><td>
<table>
<tr><th>Activity type</th><td>PRALAB</td><td>TEORIA</td></tr>
<tr><th>Number of credits</th><td>3</td><td>3</td></tr>
<tr><th>Number of groups</th><td>4</td><td>2</td></tr>
</table>
</td></tr>
<tr><td>Coordination</td><td>GIMENO ILLA, JUAN MANUEL</td></tr>
<tr><td>Department</td><td>COMPUTER SCIENCE AND INDUSTRIAL ENGINEERING</td></tr>
<tr><td>Teaching load distribution between lectures and independent student work</td><td>20% on-site
20% virtual
60% autonomous work</td></tr>
<tr><td>Important information on data processing</td><td>Consult this link for more information.</td></tr>
<tr><td>Language</td><td>Preferably Catalan (Spanish if any student shows dificulties with Catalan).</td></tr>
<tr><td>Distribution of credits</td><td>Xavier Domingo (6)
Juan Manuel Gimeno (9)
Joan Palau (3)</td></tr>
</table>

INTRODUCTION TO PROGRAMMING II 2022-23

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
DOMINGO ALBIN, JAVIER JUAN	xavier.domingo@udl.cat	6	By appointment
GIMENO ILLA, JUAN MANUEL	juanmanuel.gimeno@udl.cat	9	By appointment
PALAU ONCINS, JOAN	joan.palau@udl.cat	3	

Subject's extra information

We assume the students have all the concepts of Introduction to Programming I as we buiid upon them into two directions: object-oriented programming and recursive design.

Learning objectives

- To apply the Object Oriented Programming paradigmn to simple problems.
- To use the basic Java file types
- To design simple recursive algorithms
- To use the Java standard documentation
- To use an Intengrated Development Environment

Competences

- **Cross-disciplinary competences**
  - **EPS1:** Capacity to solve problems and prepare and defence arguments inside the area of studies.
  - **EPS5:** Capacity of abstraction and of critical, logical and mathematical thinking.
  - **EPS9:** Capacity for unidisciplinary and multidisciplinary teamwork.
  - **EPS12:** To be motivated for the quality and steady improvement.
- **Specific competences**
  - **GII-FB3:** Capacity to understand and master the basic concepts of discreet mathematics, logical, algorithmic and computational complexity, and its application to solve engineering problems.
  - **GII-FB4:** Basic knowledge of the use and programming of computers, operating systems, databases and computer programs with applications in engineering.
  - **GII-FB5:** Knowledge of the structure, organisation, operation and interconnection of the computer systems, the basics of programming, and its application to solve engineering problems.
  - **GII-FB7:** Knowledge, design and efficient use of the types and data structure more suitable for solving a problem.
  - **GII-FB9:** Capacity to know, comprise and evaluate the structure and architecture of computers, as well as the basic components that conform them.

Subject contents

INTRODUCTION TO PROGRAMMING II 2022-23

1. Introduction
    1.1 From C to Java
    1.2 The ACM Task Force Library
    1.3 The main program
    1.4 Using auxiliar functions
    1.5 Arrays in Java
    1.6 Strings in Java
2. Object Oriented Programming
    2.1 Objects and references
    2.2 Graphic classes in the ACM library
    2.3 The String class
    2.4 Class definition in Java
3. File processing
    3.1 Types of files
    3.2 Sequential text files
    3.3 Random access binary files
    3.4 MergeSort
4. Recursive design
    4.1 Function calls
    4.2 Thinking recursively
    4.3 Recursivity using cursors
    4.4 Binary search
    4.5 Multiple recursion

Software / languages / libraries:

- Java OpenJdk
- IntelliJ IDEA Community Edition
- ACM Java Task Force
- JUnit 5

Methodology

Big Size Groups: Theory Classes (3 cretits)

- Theory: Classes supportted by handnotes
- Practical application: always working on concrete examples.

Mid Size Groups: Laboratory Classes (3 credits)

- Aimed to the resolution of practical cases by the students (there is a problems collection which includes

exams from previous years)
- Personal tutoring of projects and difficulties.
- Use of an Integrated Development Environment.

Autonomous Work

- Software projects are done non-presentially.
- We recommend students to solve the problems in the collection to practice and get feedback from the teaching staff.

Development plan

Week	Big Size Group	Mid Size Group	Autonomous Work
1	Presentation + From C to Java (1 to 3)	Netbeans	Study and problem solving
2	From C to Java (rest)	Probs 3, 4 i 6	Study and problem solving
3	Introduction to OOP (1 & 2)	Probs 1, 2, 5	Study and problem solving Project 1
4	Introduction to OOP (3 & 4)	Probs 1, 2	Study and problem solving Project 1
5	Introduction to OOP (5, 6 & 7)	Probs 2, 4, 5	Study and problem solving Project 1
6	OOP Ampliation (8 & 9)	Probs 8, 9	Study and problem solving Project 2
7	OOP Ampliation (10 & 11)	Probs 10, 11, 12	Study and problem solving Project 2
8	OOP Ampliation (12 to 14)	Previous exams	Study and problem solving
9	Evaluation		
10	File management in Java (1 to 3)	Javadoc	Project 2
11	File management in Java (4 to 6)	Probs 2, 3, 4	Study and problem solving Project 2
12	File management in Java (7 & 8)	Probs 5, 6, 7	Study and problem solving Project 3
13	Recursive design (1 to 3)	Probs 8, 9 10	Study and problem solving Project 3
14	Recursive design (4 to 6)	Probs 1 i 2	Study and problem solving Project 3
15	Recursive design (9 & 10)	Probs 3, 4, 5 Previous exams	Study and problem solving
16	Evaluation		
17	Evaluation		
18	Tutories		Study and problem solving Project 3
19	Evaluation		

- Numbers in the second column correspond to the section in the handouts of the subject.
- Those in the third to the numbers in the associated problems collection.

Evaluation

Acr	Activity	Weight	Minimum grade to pass	Mandatory	Recoverable
Ex1	1st Midterm	20%	NO	NO	YES (its grade is only taken into account when it is greater that the second midterm)
Ex2	2nd Midterm	30%	4,0	YES	YES
Proj1	Project 1	15%	NO	NO	NO
Proj2	Project 2	20%	NO	NO	NO
Proj3	Project 3	15%	NO	NO	NO

Final Grade = 0,25 * Ex1 + 0,25 * Ex2 + 0,15 * Proj1 + 0,20 * Proj2 + 0,15 * Proj3

- Subject is passed if Final Grade is greater than or equal than 5
- First midterm grade is only taken into account if its greater than 2nd midterm (if not, the 2nd midterm grade is used)
- If the grade of the second midterm (or its recovery) is lower than 4, this will be the final grade, and the grades of the practices won't be taken into account.

Bibliography

- Basic:
  - Handnotes (in spanish).
  - Eric S. Roberts, The Art & Science of Java: An Introduction to Computer Science, PearsonEducation, 2008. (hay una versión preliminar disponible en pdf).
  - Eric S. Roberts, Thinking Recuersively with Java, John Wiley & Sons, 2006.
- Additional:
  - ACM Java Task Force Library Documentation http://jtf.acm.org/
  - Kathy Sierra y Bert Bates, Head First Java,O'Reilly, 2003.
  - Jorge A. Villalobos y Rubby Casallas, Fundamentos de Programación. Aprendizaje Activo Basado en Casos. Pearson Pentice-Hall, 2006