



Universitat de Lleida

DEGREE CURRICULUM  
**INTRODUCTION TO  
PROGRAMMING II**

Coordination: GIMENO ILLA, JUAN MANUEL

Academic year 2016-17

Subject's general information

<b>Subject name</b>	INTRODUCTION TO PROGRAMMING II			
<b>Code</b>	102001			
<b>Semester</b>	2nd Q(SEMESTER) CONTINUED EVALUATION			
<b>Typology</b>	<b>Degree</b>	<b>Course</b>	<b>Typology</b>	<b>Modality</b>
	Double bachelor's degree: Degree in Computer Engineering and Degree in Business Administration and Management	1	COMMON	Attendance-based
	Bachelor's Degree in Computer Engineering	1	COMMON	Attendance-based
<b>ECTS credits</b>	6			
<b>Groups</b>	2GG,4GM			
<b>Theoretical credits</b>	3			
<b>Practical credits</b>	3			
<b>Coordination</b>	GIMENO ILLA, JUAN MANUEL			
<b>Department</b>	INFORMATICA I ENGINYERIA INDUSTRIAL			
<b>Teaching load distribution between lectures and independent student work</b>	40% lectures 60% autonomous work			
<b>Important information on data processing</b>	Consult <a href="#">this link</a> for more information.			
<b>Language</b>	Preferably Catalan (Spanish if any student shows difficulties with Catalan).			
<b>Distribution of credits</b>	Xavier Domingo (6) Juan Manuel Gimeno (9) Toni Granollers (3)			
<b>Office and hour of attention</b>	Xavier Domingo (1.06 EPS by appointment). Juan Manuel Gimeno (wednesday at 1pm at office 3.20 EPS; others by appointment). Toni Granollers (3.10 EPS by appointment).			

Professor/a (s/es)	Adreça electrònica professor/a (s/es)	Crèdits	Horari de tutoria/lloc
DOMINGO ALBIN, JAVIER JUAN	xdomingo@diei.udl.cat	6	By appointment.
GIMENO ILLA, JUAN MANUEL	jmgimeno@diei.udl.cat	9	By appointment.
GRANOLLERS SALTIVERI, ANTONI	antoni.granollers@udl.cat	3	By appointment.

## Subject's extra information

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

## Learning objectives

- To apply the Object Oriented Programming paradigm to simple problems.
- To use the basic Java file types
- To design simple recursive algorithms
- To use the Java standard documentation
- To use an Integrated 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 discrete 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

1. Introduction

1.1 From C to Java

1.2 The ACM Task Force Library

1.3 The main program

1.4 Using auxiliary 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

## Methodology

### **Big Size Groups: Theory Classes (3 credits)**

- Theory: Classes supported 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	25%	4,0	YES	YES (with 2nd midterm)
Ex2	2nd Midterm	25%	4,0	YES	YES (with recovery exam with a max grade of 7)
Proj1	Project 1	15%	NO	NO	YES (at week 18 with a max grade of 5)
Proj2	Project 2	20%	NO	NO	YES (at week 18 with a max grade of 5)
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 or equal than 5
- A passed 2nd midterm recovers a failed 2nd midterm
- First midterm grade is only taken into account if its greater than 2nd midterm (if not, the 2nd midterm grade is used)
- If the student has to take the recovery exam, the maximum grade of theory is 7
- If the student has to re-send the first or second project at the end of the course, the project will get a maximum grade of 5.

## 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