



DEGREE CURRICULUM

# INTRODUCTION TO PROGRAMMING II

Coordination: MARTINEZ RODRIGUEZ, SANTIAGO

Academic year 2023-24

## Subject's general information

Subject name	INTRODUCTION TO PROGRAMMING II			
Code	105001			
Semester	2nd Q(SEMESTER) CONTINUED EVALUATION			
Typology	Degree	Course	Character	Modality
	Bachelor's Degree in Computer Engineering	1	COMMON/CORE	Attendance-based
Course number of credits (ECTS)	6			
Type of activity, credits, and groups	Activity type	PRALAB		TEORIA
	Number of credits	3		3
	Number of groups	2		1
Coordination	MARTINEZ RODRIGUEZ, SANTIAGO			
Department	COMPUTER ENGINEERING AND DIGITAL DESIGN			
Teaching load distribution between lectures and independent student work	6 ECTS = 25x6 = 150 working hours: 40% -> 60 in-class hours, 60% -> 90 autonomous work hours.			
Important information on data processing	Consult <a href="#">this link</a> for more information.			
Language	Catalan.			
Distribution of credits	Theory: 3 Practices: 3			

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
LOPEZ FERNANDEZ, JESUS MARIA	jesus.lopez@udl.cat	6	Arrange with the teacher. Optionally, by videoconference.
MARTINEZ RODRIGUEZ, SANTIAGO	santi.martinez@udl.cat	3	Arrange with the teacher. Optionally, by videoconference.

## 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

The main learning objectives are:

- 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 / Module of basic training

- **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.

### Specific Competences / Module of common training in the computer branch

- **GII-CRI7.** Knowledge, design and efficient use of the types and data structure more suitable for solving a problem.
- **GII-CRI9.** Capacity to know, comprise and evaluate the structure and architecture of computers, as well as the basic components that conform them.

## Subject contents

### Unit 1. Introduction to Java

- 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

## Unit 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

## Unit 3. File processing

- 3.1 Types of files
- 3.2 Sequential text files
- 3.3 Random access binary files
- 3.4 MergeSort

## Unit 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

Each week students attend 2 hours with a Large Group and 2 hours with a Medium Group. Medium Group sessions are practices.

### Large Group: Theory Classes (3 ECTS)

- Theory: Classes supported by class notes.
- Practical application: always working on concrete examples.

### Medium Group: Practical Classes (3 ECTS)

- Aimed to the resolution of practical cases by the students (there is a collection of problems).
- Personal tutoring of projects and difficulties.

### 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	Large Group Activity	Medium Group Activity	Autonomous Work
1	Presentation + From C to Java	javac and java. Problem 1, 2	Study and problem solving
2	From C to Java (cont.)	Problems 3, 4, 5, 6, 7	Study and problem solving
3	Introduction to OOP	Problems 1, 2	Study and problem solving Project 1
4	Introduction to OOP	Problems 3, 4	Study and problem solving Project 1
5	Introduction to OOP	Problems 5, 6	Study and problem solving Project 1
6	Further OOP	Problems 7, 8	Study and problem solving Project 1
7	Further OOP	Problems 9, 10, 11	Study and problem solving Project 2

# INTRODUCTION TO PROGRAMMING II 2023-24

Week	Large Group Activity	Medium Group Activity	Autonomous Work
8	Further OOP	Previous exams	Study and problem solving Project 2
9	Evaluation		
10	File management	Sol. exam. Problem 1	Project 2
11	File management	Problems 2, 3, 4	Study and problem solving Project 2
12	File management	Problems 5, 6, 7	Study and problem solving Project 3
13	File management	Problems 8, 9, 10	Study and problem solving Project 3
14	Recursive design	Problems 1, 2	Study and problem solving Project 3
15	Recursive design	Previous exams	Study and problem solving Project 3
16	Evaluation		
17	Evaluation		
18			Study and problem solving
19	Recovery		

## Evaluation

### Continuous evaluation

Block	Evaluation Activity	Weight	Minimum Score	Group	Recoverable
EP1	1st Midterm Exam	20%	No	No	Yes (with 2nd midterm exam)
EP2	2nd Midterm Exam	30%	4	No	Yes
PR1	Practice 1	15%	No	Yes ( $\leq 2$ )	No
PR2	Practice 2	20%	No	Yes ( $\leq 2$ )	No
PR3	Practice 3	15%	No	Yes ( $\leq 2$ )	No
To pass the subject the final score must be $\geq 5$ and the minimum marks indicated must be achieved.					
<b>Final Score</b> = $0.2 \cdot EP1 + 0.3 \cdot EP2 + 0.15 \cdot PR1 + 0.2 \cdot PR2 + 0.15 \cdot PR3$					

### Remarks:

- 1st midterm exam score is only taken into account if it's greater than 2nd midterm exam (if not, the 2nd midterm exam score is used).

### Alternative evaluation (students who waive continuous evaluation):

Students who have the approval to be evaluated by alternative evaluation (see requirements and procedure in the evaluation regulations) will have to do the following activities.

- Exam (80%):** It can be improved. A minimum grade of 4 is required. Date of the exam: the date of the realization of the 2nd Midterm Exam, defined by the EPS.
- Practice (20%):** It cannot be improved. A minimum grade is not required. Delivery date: the date of the realization of the 2nd Midterm Exam, defined by the EPS.
- Improvement of exam (80%):** A minimum grade of 4 is required. Date of the exam: the date of the realization of the Improvement Exam, defined by the EPS. The realization of Improvement of Theory exam does not condition the maximum grade achieved in the subject.

## Bibliography

### Basic

- Class notes (in spanish).
- Eric S. Roberts: The Art & Science of Java: An Introduction to Computer Science, Pearson Education, 2008 (there is a preliminary version available in pdf).
- Eric S. Roberts: Thinking Recuersively with Java, John Wiley & Sons, 2006.

### Complementary

- Documentation of the ACM Java Task Force library <http://jtf.acm.org/>
- Kathy Sierra, Bert Bates: Head First Java, O'Reilly, 2003.
- Jorge A. Villalobos, Rubby Casallas: Fundamentos de Programación. Aprendizaje Activo Basado en Casos. Pearson Pentice-Hall, 2006.