



DEGREE CURRICULUM

OBJECT ORIENTED PROGRAMMING

Coordination: MARTINEZ RODRIGUEZ, SANTIAGO

Academic year 2023-24

Subject's general information

Subject name	OBJECT ORIENTED PROGRAMMING			
Code	102368			
Semester	2nd Q(SEMESTER) CONTINUED EVALUATION			
Typology	Degree	Course	Character	Modality
	Bachelor's degree in Digital Interaction and Computing Techniques	1	COMMON/CORE	Attendance-based
Course number of credits (ECTS)	6			
Type of activity, credits, and groups	Only examination			
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. There are no classes for this subject.			
Important information on data processing	Consult this link for more information.			
Language	Catalan.			
Distribution of credits	There are no classes for this subject.			

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
MARTINEZ RODRIGUEZ, SANTIAGO	santi.martinez@udl.cat	0	Arrange with the teacher. Optionally, by videoconference.

Subject's extra information

We assume the students have all the concepts of Algorithms and Programming 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

Basic Competences

- **B01.** That students have demonstrated to possess and understand knowledge in an area of study that starts from the base of general secondary education, and is usually found at a level that, although supported by advanced textbooks, also includes some aspects that imply knowledge coming from the vanguard of his/her field of study.

Transversal Competences

- **CT3.** Acquire training in the use of new technologies and information and communication technologies.
- **CT5.** Acquire essential notions of scientific thought.

General Competences

- **CG2.** Design, develop, evaluate and guarantee the accessibility, ergonomics, usability and security of computer systems.
- **CG3.** Use adequate hardware and software platforms to develop and execute interactive digital applications.
- **CG5.** Know the basic subject areas and technologies needed to learn and develop new methods and technologies, and those that help to adapt to new situations.
- **CG7.** Solve problems through initiative, determination, independence and creativity.
- **CG8.** Capacity for abstraction and critical, logical and mathematical reasoning.

Specific Competences

- **CE2.** Capacity to understand and master the basic concepts of discrete mathematics, logics, algorithmic and computational complexity, and its application to solve computational problems.
- **CE3.** Basic knowledge of the use and programming of computers, operating systems and databases, and their use in the development of interactive applications.
- **CE4.** Capacity to know, understand and evaluate the structure and architecture of computers, as well as the basic components that conform them.
- **CE16.** Capacity to design and evaluate person-computer interfaces that guarantee the usability of systems, services and computer applications.
- **CE17.** Capacity to apply knowledge on design to propose and defend a design concept for an interactive system and use proper creative technologies to develop each project.
- **CE24.** Capacity to understand the human factors involved in any interactive process between humans and technology, as well as being able to adequately apply them in the design of interactive products and services, and their interfaces.

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

There are no classes for this subject.

However, the student who wishes may attend, as an auditor, the equivalent subject (Introduction to Programming II) of the Degree in Computer Engineering that is taught on the same Igualada campus.

Development plan

There are no classes for this subject.

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 (≤ 2)	No
PR2	Practice 2	20%	No	Yes (≤ 2)	No
PR3	Practice 3	15%	No	Yes (≤ 2)	No
To pass the subject the final score must be ≥ 5 and the minimum marks indicated must be achieved.					
Final Score = $0.2 \cdot EP1 + 0.3 \cdot EP2 + 0.15 \cdot PR1 + 0.2 \cdot PR2 + 0.15 \cdot PR3$					

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 Recursively 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.