



ACADEMIC PROGRAM

OBJECT ORIENTED PROGRAMMING

B.F.A. IN COMPUTER SCIENCE

MODALITY: ON CAMPUS

ACADEMIC YEAR: 2022-2023

Name of the course:	Object Oriented Programming
Degree :	Computer Science
Location:	Centro Universitario de Tecnología y Arte Digital
Area:	Programming
Year:	2º
Teaching period:	1
Type:	OB
ECTS credits:	6
Teaching modality:	On campus
Language:	English
Lecturer / Email	-
Web page:	http://www.u-tad.com/

SUBJECT DESCRIPTION

Area description

This subject belongs to the programming subject. This subject is dedicated to the study of programming techniques and languages on which the software engineering degree studies will be based.

Subject description

This subject allows the student to acquire key knowledge about the object-oriented programming paradigm, create programs structured in classes and equipped with mechanisms such as inheritance and polymorphism, as well as work with an object-oriented programming language.

COMPETENCIES AND LEARNING OUTCOMES

Competencies

BASIC AND GENERAL SKILLSs

CG1 - Ability to understand, schedule and solve problems trough software development

CG3 - Knowledge of the scientific fundamentals applicable to the resolution of computer problems

CG4 - Ability to simplify and optimize computer systems by understanding their complexity

CG9 - Ability to learn, modify and develop new software solutions

CG10 - Use of creative techniques to carry out computer projects

CB1 That students have demonstrated knowledge and understanding in an area of study that starts from the basis of general secondary education, and is usually at a level that, although it is supported by advanced textbooks, also includes some aspects that involve knowledge from the forefront of their field of study.

CB2 Students are able to apply their knowledge to their work or vocation in a professional manner and possess the competences usually demonstrated through the development and defense of arguments and problem solving within their field of study.

CB3 That students have the ability to gather and interpret relevant data (usually within their area of study) in order to make judgements that include reflection on relevant social, scientific or ethical issues.

CB4 Students are able to convey information, ideas, problems and solutions to both specialist and non-specialist audiences.

CB5 That students have developed those learning skills necessary to undertake further studies with a high degree of autonomy.

SPECIFIC SKILLS

CE1 - Knowledge of the structure of computers, the concepts of coding, manipulation, information processing and use of low-level languages

CE7 - Knowledge of the main types of data structures and use of libraries and algorithmic techniques associated with these structures together with the complexity orders that characterize these techniques

CE8 - Conocimiento de los distintos paradigmas detrás de los lenguajes de programación/ Knowledge of the different software

paradigms that underpin programming languages

CE9 - Knowledge of effective control structures, variables, programming syntax and memory management in the development of a computer application

CE10 - Ability to work with a release manager and generate application documentation automatically.

CE15 - Knowledge of fault tolerance, adaptability, load balancing and system predictability for distributed application development

CE17 - Knowledge of the parallelization characteristics of graphics cards and high-performance architectures for application development.

CE20 - Ability to test the operation and functionality of a computer application, develop test plans and use test-oriented design and programming techniques

CE23 - Knowledge of the principles of artificial intelligence and use of deterministic search algorithms and state machines

Learning outcomes

Upon completion of the degree, the graduate will be able to:

- To understand and handle the concept of dynamic memory

- To identify classes with the relevant data of a problem
- To instance classes and objects and manage them
- To understand and use the mechanisms of inheritance, polymorphism and operator overloading
- To identify class relationships in different use-cases.
- To master an object oriented programming language.
- To master programming patterns
- To know different problem solution strategies from an algorithmic view point: divide and conquer, dynamic programming, backtracking or genetic algorithms.
- To understand algorithmic complexity, assess it and search for optimal solutions
- To code a program able to find the optimal path between any pair of nodes of a graph
- To build neural networks to solve applied problems

CONTENTS

Object Oriented Programming concepts

Exception handling

SUBJECT SYLLABUS

Topic 1: Introduction to Object-Oriented Programming

Procedural Programming vs. Object-oriented programming.

Objects and classes.

Properties of object orientation.

Topic 2: Programming in Java language

Development environment.o Operators.

Branches.or Loops.or Variables.or Arrays.

Argument lists.

Topic 3: Classes in Java

Class concept.

Objects and methods.

Member variables.

Builders.

Scope of the variables.

Using this.

Packages.

Composition relations.

Topic 4: Inheritance or inheritance hierarchies.

Object conversion.

Polymorphism.

Abstract classes or Interfaces.

Topic 5: Exception Handling

Standard exceptions in Java.

Throwing and catching exceptions.

Creating exceptions.

Topic 6: GUI programming

Windows.

Buttons.

Event management.

Graphic text components.

Disposition managers.

Topic 7: Data input and output in Java

Reading and writing classes.

Standard input and output.

The Scanner class.

Text files.

Binary files.

Topic 8: Templates and generic programming.

Generic classes.

Generic methods

TRAINING ACTIVITIES AND TEACHING METHODOLOGIES

TRAINING ACTIVITIES

LEARNING ACTIVITIES	Total hours	Hours of presence
<i>Theoretical / Expository classes</i>	35,64	35,64

<i>Practical classes</i>	18,91	18,91
<i>Tutorials</i>	4,00	2,00
<i>Independent study and autonomous work of the student</i>	51,82	0,00
<i>Elaboration of work (group or individual)</i>	33,82	0,00
<i>Evaluation Activities</i>	5,82	5,82
TOTAL	150	62,37

Teaching methodologies

Expository method or master lesson

Case learning

Learning based on problem solving

Cooperative or collaborative learning

inquiry learning

Flipped classroom methodology

Gamification

Just in time Teaching (JITT) or classroom on time

Expository method or master lesson

Case method

Learning based on problem solving

Cooperative or collaborative learning

inquiry learning

Flipped classroom methodology

Gamification

TEMPORAL DEVELOPMENT

DIDACTIC UNITS / TOPICS TIME PERIOD

Topic 1: Introduction to Object-Oriented Programming1 week

Topic 2: Programming in Java language 2 weeks

Topic 3: Java classes 3 weeks

Topic 4: Inheritance 3 weeks

Topic 5: Exception Handling 1 week

Topic 6: GUI programming 2 week

Topic 7: Data input and output in Java 2 week

Topic 8: Templates and generic programming. 1 week

EVALUATION SYSTEM

ASSESSMENT SYSTEM	MINIMUM SCORE RESPECT TO THE FINAL ASSESSMENT (%)	MAXIMUM SCORE RESPECT TO THE FINAL ASSESSMENT (%)
<i>Assessment of participation in class, exercises or projects of the course</i>	0	30
<i>Assessment of assignments, projects, reports, memos</i>	30	80
<i>Objective test</i>	10	60

GRADING CRITERIA

ASSESSMENT SYSTEM	ORDINARY EVALUATION	EXTRAORDINARY EVALUATION
<i>Assessment of participation in class, exercises or projects of the course</i>	0	0
<i>Assessment of assignments, projects, reports, memos</i>	30	30
<i>Objective test</i>	70	70

General comments on the evaluations/assessments

Throughout the course, a project is proposed that must be delivered on the indicated date through the virtual platform. This work will be evaluated with a demonstration and will account for 30% of the grade. Late deliveries are not accepted, and if they are accepted for justified cause, the delay will mean a considerable reduction in the grade.

- Practices will be proposed that represent a maximum of 10% of the final grade.
- The final grade in continuous evaluation is, therefore, the following sum: $0.30 \cdot \text{project} + 0.10 \cdot \text{practices} + 0.6 \cdot \text{objective test}$.
- To pass the subject in the ordinary call, it is essential that the final grade is at least 5.0 (out of 10), a minimum grade of 4 in the project or objective test is necessary to be compensable.
- Ordinary call: Those students who do not exceed the grade 5.0 in the previous criterion. There will be an exam for the theoretical part that will account for 70% of the final grade. If the student did not exceed the minimum grade of 4.0 in the project, she will have the opportunity to resubmit it.
- Extraordinary or single call: Students suspended in the ordinary call or students who have been granted the single call. There will be an exam for the theoretical part that will account for 70% of the final grade. If the student did not exceed the minimum grade of 4.0 in the project or did not complete it, she will have the opportunity to present it.
- No grades of any kind will be kept between different academic years.
- The use of mobile phones in the classroom is not allowed during the continuous evaluation period, unless expressly indicated otherwise by the teacher. Laptops may only be used for activities related to the subject. The teacher may withdraw the right to use the computer from those students who use it for activities that are not related to the subject (checking emails, news or social networks, consulting or preparing activities for other subjects, etc.).
- Active participation will be required from the student, necessary for the development of the classes.
- The student will be required to behave well at all times during the classes. Bad behavior that prevents the normal development of the class may lead to expulsion from the classroom for a period of time to be determined by the teacher.

LIST OF REFERENCES (BOOKS, PUBLICATIONS, WEBSITES):

Basic:

Deitel & Deitel. Java How to program.

Javier García de Jalón et. Al., Learn Java as if you were in first grade. University of Navarra.

Roger Cadenhead, Java 8. Anaya Multimedia.

Recommended:

Matt Weisfeld, The Object-Oriented Thought Process.

Addison Wesley. David Etheridge, Java: The Fundamentals of Objects and Classes – An Introduction to Java Programming. Ventus Publishing ApS.

C. Thomas Wu, Object-Oriented Programming with Java. McGraw Hill.

Kathy Sierra & Bert Bates, Head First Java. O'Reilly.

David J. Eck, Introduction to Programming Using Java. Hobart and William Smith Colleges

REQUIRED MATERIALS, SOFTWARE AND TOOLS

Type of classroom

Theory classroom

Board and projection system

Materials:

Computer with Windows or Linux

Software:

JDK de Java

Eclipse