UNIVERSIDAD DE EXTREMADURA

COURSE PROGRAM: Information and Data Structures

CODE: 501429

ACADEMIC YEAR: **2024/2025**

COURSE PROGRAM
Academic Year: 2024/2025

Identification and characteristics of the course				
Code	501429	ECTS Credits		6
Course name (English)	Information and Data Structures			
Course name (Spanish)	Estructuras de Datos y de la Información			
Degree programs	Computer Sciences in Information Technology and in Telematics Engineering			
Faculty/School	Centro Universitario de Mérida			
Semester	2nd	Type of course		Compulsory
Module	Compulsory			
Matter	Computer Engineering			
Lecturer/s				
Name		Office	E-mail	Web page
Luis Arévalo		9	ljarevalo@unex.es	
José Enrique Moguel			enrique@unex.es	
Subject Area	Languages and Computing Systems			
Department	Computer and Telematic Systems Engineering			
Coordinator (if more than one)	Luis Arévalo Rosado			

Competencies*
CE2 (**GIT**): Basic knowledge about the use of computers and computer programming, operating systems, databases and software applied to engineering.
CE7 (**GIT**): Ability in the use of communication and computer applications (office tools, databases, advanced calculators, project management, visualization, etc.) to support the development and operation on telecommunication and electronic networks, services and applications.
CE4 (**GIITI**): Basic knowledge on the use of computers and computer programming, operating systems, databases and software applied in engineering.
CE5 (**GIITI**): Knowledge of the structure, organization, operation and interconnection of computer systems, their programming fundamentals, and their application for solving engineering problems.
CE13 (**GIITI**): Knowledge, design and efficient use of the most appropriate type of data structure to the resolution of a specific problem.

Contents
Course outline*
Information and Data Structures
Course syllabus

**** The sections concerning competencies, course outline, educational activities, teaching methodologies, learning outcomes and assessment systems must conform to that included in the ANECA verified document of the degree program.

Theoretical lessons

Unit 1: Object oriented programming

- Programming introduction
- Introduction to abstract datatypes (ADTs)
- Introduction to object oriented programming
  - Objects and classes
  - Encapsulation
  - Composition, inheritance, and delegation
  - Polymorphism

Unit 2: Complexity

- Introduction to complexity
- Search and sort algorithms: complexity

Unit 3: Lineal abstract datatypes

- Introduction
- Stack datataype
  - Static and dynamic implementations
  - Use of a stack
- Queues
  - Static and dynamic implementations
  - Use of a queue
- Lists
  - Static and dynamic implementations
  - Use of a list
- Java collections

Unit 4: Functional abstract datatypes

- Introduction to functional abstract datatypes
- Sets
- Dictionaries
- Hash tables

Unit 5: Non lineal abstract datatypes

- Introduction
- AVL trees

Practical lessons

- Week 1: Introduction to java
- Week2: Arrays and Strings in Java
- Week3: Object oriented Programming in Java
- Week4: Composition
- Week5: Inheritance
- Week6: Interfaces
- Week7: Lineal ADTs - I
- Week8: Lineal ADTs - II
- Week9: Lineal ADTs - III
- Week10: Lineal ADTs - IV
- Week11: Abstract ADTs
- Week12: Non Lineal ADTs
- Week13: Non Lineal ADTs
- Week14: Non Lineal ADTs
- Week15: Practical Exam

Educational activities *								
Student workload in hours by lesson		Lectures	Practical activities				Monitoring activity	Homework
Lesson	Tota	L	HI	LAB	COM	SEM	SGT	PS

	I							
Presentation	3	1			-		-	2
1	38	6			12		1	19
2	6	2			-		-	4
3	34	5			8		1	20
4	20	5			2		1	12
5	24	8			6		-	10
Assessment **	25	3			2			**20**
TOTAL ECTS	150	30			30		3	87

L: Lectures (85 students)
HI: Hospital internships (7 students)
LAB: Laboratory or field practices (15 students)
COM: Computer room or language laboratory practices (20 students)
SEM: Problem classes or seminars or case studies (40 students)
SGT: Scheduled group tutorials (educational monitoring, ECTS type tutorials)
PS: Personal study, individual or group work and reading of bibliography

Teaching Methodologies*

- Lectures and problem-solving activities.
- Weekly practical sessions to develop programming exercises individually.
- Tutoring: follow-up of practices and exercises, resolution of doubts, and tutoring in small groups or individually.
- Guides and tips on how to deliver an oral presentation successfully.
- Self-evaluation and peer-assessment activities to develop critical thinking.
- Encourage autonomous learning by making presentations, studying the subject and solving practical case studies.

Learning outcomes *

Students should be able to:

- Know basic elements of programming (paradigms, control of structures, data structures, languages, algorithms, etc.) and apply these elements efficiently and correctly to problem solving.
- Know main linear and nonlinear data structures and how to apply them to the development of an object oriented program.

Assessment systems *

The evaluation will be continuous, based on the following criteria:

Assessment	Percentage
Exams	50,00 %
Oral presentation	0,00 %
Deliveries works (reports, practical cases, exercises and problems).	50,00 %
Attendance and / or participation in the classroom	0,00 %

***** Enter this row as many times as necessary in this table. As an example, you can set one row for partial exam and another for final exam.

OPTION A: Continuous evaluation

- Laboratory activities (LA) (15%). (Deliveries work)
The delivery of the requested activities in the laboratory will be valued.
- Activities of main group(AG) (10%). (exam work)
The delivery of different activities done in or outside the class related to the content taught in GG (Test, questionnaire, ..)
- Laboratory Deliveries (PL) (35%). (deliveries work)
There will be one or several practices that will require the development of a program by a student.
Note: During the exam period, students can either request an additional practical delivery or a practical exam to demonstrate the practice authorship.
- Final exam (EF) (40%). (exam work)
There will be a final written exam that will include both theoretical and practical contents of the subject.

Final evaluation = AL*0.15+ AG*0.1 + PL*0.35 + EF*0.40

Note:

- In the final exam, students must obtain a minimum score of 4 out of 10 to pass the exam
- Related to the final practices (PL), final delivery must run correctly, as well as students must pass their authorship exam.

OPTION B: Global Evaluation

For students who do not want to be evaluated using the continuous assessment, or they cannot for work reasons, in the final exam day there will be an additional examination to recover those parts in which the students do not have evaluation. Besides, they also have to do the Laboratory Deliveries and to pass the exam authorship.

Final evaluation = Laboratory Deliveries*0.40 + Exam*0.60

Bibliography (basic and complementary)

Recommendation Readings*

- Piensa en Java. 4ª Edición. Pearson Prentice Hall. ISBN 13: 9788489660342

Other resources and complementary educational materials

Optional Readings*

- Core Java 2 Vol I. Fundamentos. Pearson Prentice Hall/Sun. ISBN 13:9788420548326
- Core Java 2. Vol II. Características Avanzadas. Pearson Prentice Hall/Sun.ISBN 13: 9788483223109
- Programación, Algoritmos y ejecicios resultos en Java. Pearson Prentice Hall. ISBN 13: 9788420540245
- Estructuras de datos con Java. Diseño de estructuras y algoritmos. Pearson Addison Wesley. ISBN 13: 9788420550343

Online Readings*

Aprenda Java como si estuviera en primero. Manual en PDF.