Universidad Zaragoza

Academic Year/course: 2023/24

30218 - Programming Theory

Syllabus Information

Academic year: 2023/24
Subject: 30218 - Programming Theory
Faculty / School: 110 - Escuela de Ingeniería y Arquitectura
326 - Escuela Universitaria Politécnica de Teruel
Degree: 439 - Bachelor's Degree in Informatics Engineering
443 - Bachelor's Degree in Informatics Engineering
ECTS: 6.0
Year: 2
Semester: Second semester
Subject type: Compulsory
Module:

1. General information

The objective of this subject is for the student to consolidate the programming knowledge acquired in previous courses and to learn more advanced concepts that appear in the main programming paradigms.

Object Oriented Programming will be studied, starting from the theoretical concepts involved and a set of problems present in current software development, extending the programming techniques acquired in previous subjects.

Functional Programming will be presented within a new theoretical framework for the student, introducing its basic techniques.

These approaches and objectives are aligned with the Sustainable Development Goals (SDGs) of the United Nations Agenda 2030 (https://www.un.org/sustainabledevelopment/es/), insofar as the acquisition of the learning results of the subject provides training and competence to contribute to their achievement, by building more energy efficient software and improving the technological capacity of industrial sectors (in particular, goals 7.3, 8.2, 9.5 and 9.c).

2. Learning results

Upon completion of the subject, the student will be able to:

- **R12**. Develop object-oriented programs that incorporate graphical user interfaces, manage events or can access databases and distributed network resources.
- **R13**. Know and understand the syntax and semantics of a functional programming language.
- **R14**. Develop programs written in a functional language.
- **R15**. Have a perspective of other programming paradigms and languages.

Must know the basic algorithmic procedures of computer technologies to design and implement solutions to problems, analyzing the appropriateness and complexity of the proposed algorithms, in a robust and efficient, choosing the most appropriate paradigm and programming languages, and apply these techniques to software development.

The knowledge acquired complements and extends what has already been learned in previous subjects of the programming block, and will provide the student with a global perspective of the programming technologies and their use in different contexts.

3. Syllabus

Block 1. Object Oriented Programming

- Classes
- Polymorphism and Inheritance
- Generic Programming
- Class hierarchies and genericity
- Containers and Data Structures
- Robust application design

- Type inference
- Design patterns
- Access to network resources

Block 2: Functional Programming

- Introduction to Functional Programming
- Functional languages: basic types
- Expressions and Functions: recursion
- Lists and higher-order functions
- Algebraic types and classes
- Functional programming in object-oriented languages

Block 3. Other paradigms

- Logical Paradigm
- Dynamic languages

4. Academic activities

The student's dedication to achieve the learning results in this subject is estimated in 150 hours, distributed as follows:

- 60h of face-to-face activities (lectures, problems and practicals)
- 25h of team programming work
- 60h of individual personal work
- 5h of theory and practical final exams

Face-to-face activities are organized as follows:

- Theoretical classes: 30h (2 hours per week)
- Problem solving and case studies: 15h (1 hour per week)
- Practical laboratory classes: 15h (2-hour sessions, one session every two weeks)

They can be done in groups of a maximum of 2 people.

Individual student personal work activities may include:

- The resolution of certain exercises that are proposed as work prior to the problem sessions.
- A major final programming project to be carried out in a team with other students.

5. Assessment system

The student must demonstrate that they have achieved the intended learning results through the following assessment activities:

- **Final theory test:** written exam in which the student must solve programming exercises and, if necessary, answer conceptual questions conceptual questions.
- **Assessment of the practical activities** carried out during the term:
  - **Laboratory practices, programming project and resolution of proposed exercises:** with the practical programming works, the student's work and learning progress will be monitored.
    Students who have met the deadlines and have demonstrated an adequate level of achievement and quality will be exempted from the practical programming exam.
  - **Practical programming exam:** individual exam in which the student will have to solve programming exercises similar to those done in the practical or problem sessions. Students who do not pass the assessment of the practical part of the subject within the deadlines specified throughout the term will have to

take this exam Students who are exempted from taking this exam and choose to take it, will waive the grade already obtained with the delivery of their practical work.

The weighting of the assessment activities is as follows:

	Theory exam	Practical activities
School of Engineering and Architecture, Ebro River Campus	60%	40%
Polytechnic University School, Teruel Campus	40%	60%

It is necessary to pass the theory exam with a minimum grade of 5.0 points to pass the subject. In this case, the student's final grade in the subject is obtained as the weighted sum of the grades of the theoretical and practical parts with their corresponding weights . If the grade in the theory exam is lower than 5.0, the grade in the course will be the grade obtained in that exam.