

Academic Year of Degree: 2025/26

24294 - Functional Design of Programs

Teaching Guide Information

Academic Course: 2025/26

Academic Center: 337 - Engineering School

Study: 3377 - Bachelor's Degree in Computer Engineering

Subject: 24294 - Functional Design of Programs

Credits: 5.0

Course: 634 - Bachelor's degree in Computer Engineering: 2
702 - Minor in Programming: 1

Teaching languages:

Theory: Group 1: English

Group 2: Spanish

Practice: Group 101: English

Group 201: Spanish

Seminar: Group 201: Spanish

Teachers: Rafael Ramirez Melendez

Teaching Period: First term

Presentation

Functional Design of Programs is a compulsory course presented during the computer engineering degree. As one of the courses in the field of Programming Languages, it is concerned with the presentation of concepts and logical constructs of functional languages as a set of tools and alternative mindset for the design of algorithms and development of software systems.

The course is structured with theory, practice, and seminar classes. In theory classes, concepts are introduced and examples of their applications are presented. In the seminar classes, students can consolidate their knowledge by working on small-scale problems related to the theory. Finally, in practice sessions, problems of higher complexity are proposed for the application of learned concepts directly in a programming environment.

Associated skills | General learning outcomes

Basic Competences

- CB2. That the students can apply their knowledge to their work or vocation in a professional way and possess the competences which are usually proved by means of the elaboration and defense of arguments and the solution of problems within their study area;
- CB5. That the students have developed those learning abilities necessary to undertake later studies with a high degree of autonomy.

Transversal Competences

- CT3. Applying with flexibility and creativity the acquired knowledge and adapting it to new contexts and situations;

- CT5. Generating new ideas and incorporating them in the daily work.

Specific Competences

- CE8. Mastering the concepts of programming and data structures, including principles of secure design and defensive programming, program verification and error detection;
- CE10. Recognizing basic algorithmic procedures and applying them for the resolution of computational problems, analyzing the solution's suitability and complexity.

Learning outcomes | Specific learning outcomes

Specific Learning Outcomes

- RA.CE8.2 Design and use the adequate types and data structures for solving a problem;
- RA.CE8.3 Design and use advanced data structures and the most proper suitable algorithms for solving a problem.
- RA.CE8.4 Apply functional design programming techniques;
- RA.CE10.1 Solve and analyze computational problems applying common algorithmic strategies.

Contents

1. Concepts of programming languages

- Programming paradigms
- Attributes of a programming language
- Syntax and semantics of programming languages
- Implementation of programming languages
- Program verification

2. Elements of Logical Languages

- First-order logic
- Horn clauses
- predicates
- unification
- deduction
- Backtracking

3. Elements of Functional Languages:

- Functions as First-Class Entities
- Side-Effects-Free Programming
- Pattern Matching
- Higher-order Operations

4. Functional Design Concepts

- Function closures
- Type-Inference and Polymorphism
- Lazy Evaluation
- Memoization

Sustainable Development Goals

4 - Quality Education

9 - Industry, Innovation and Infrastructure

Evaluation and grading system

In the continuous assessment, each of the three activities that make up the subject is taken into account: theory classes, laboratories and seminars:

Activitat	Avaluació	Recuperable?
Teoria	Avaluació mitjançant un examen final	Si
Laboratoris	Avaluació amb les pràctiques de programació (LP) i examen final de practiques (LE)	LP: No LE: Si
Seminaris	Avaluació mitjançant resolució d'exercicis de seminari	No

T: the evaluation of the theory through a final exam (retrievable in July)

L: the evaluation of the laboratories with the programming practices (not recoverable) and final practical exam (recoverable in July).

S: the evaluation of the seminars (not recoverable)

In the assessment, each of the three activities must be passed separately and the final grade is obtained by taking the weighted average as follows:

$$\text{Nota Final} = 0,5 * T + 0,4 * L + 0,1 * S$$

The theory exam will be based on the content developed in the theory class and in the seminars. It is a written and individual exam that assesses all the skills developed throughout the course. This assessment is mandatory and must be graded with at least 50% in order to pass the subject.

In the laboratories, a series of practices are carried out that test the student's ability to apply the theory in the form of programs on a computer. Practices are carried out in pairs, so students must cooperate and know how to communicate to solve problems. The evaluation of the practices is also mandatory and is graded in 2 parts: a continuous evaluation throughout the term in the form of delivery of computer programs (LP), and the other as part of the final exam of the subject dedicated to practice subjects (LE). Each of the parts must be qualified with at least 50% to pass the subject. The grade L corresponding to the practice grade is obtained by taking the weighted average as follows:

$$L = 0,6 * LP + 0,4 * LE$$

The details of the assignment submission deadlines throughout the quarter and the penalties for late delivery will be published on the subject's global campus.

Before each seminar, problems will be presented to the students so that they can solve them in teams of three or four before the session, as preparation for the seminar. These problems correspond to concepts or knowledge discussed in theory class. The students will have time to finalize their answers to the problems at the beginning of the seminar, after which they will randomly weigh their solutions on the board. The evaluation will consider the seminar's preparation, the presentation of the proposed solution, and the use of the seminar. The evaluation of the seminar work is mandatory.

In case of not passing the subject in the assessment set out in the previous points, the student has the right to a call in the month of July. In the July call, it will only be possible to recover part of the theory exam and part of the final practice exam. In other words, the part of programming practices and the part of seminars are not recoverable in the July call. The mark for this call will be obtained in the same way as the mark for the ordinary call explained above.