Universidad Zaragoza

Academic Year/course: 2021/22

30007 - Fundamentals of computer studies

Syllabus Information

Academic Year: 2021/22
Subject: 30007 - Fundamentos de informática
Faculty / School: 110 - Escuela de Ingeniería y Arquitectura
Degree: 436 - Bachelor's Degree in Industrial Engineering Technology
ECTS: 6.0
Year: 1
Semester: 436-First semester o Second semester
107-First semester
Subject Type: Basic Education
Module:

1. General information

1.1. Aims of the course

This course aims to:

1. Enable the student to plan and design the solution to a problem based on simple programs. Consequently, its core content is programming, and particularly, the specification of problems, the design of alternate solutions, the choice of the best algorithm based on experimentation and the transformation of such solutions in executable software in the computer in a general purpose language.
2. Facilitate the student's knowledge of basic computer hardware elements, its basic inner workings and that the student is able to search for solutions autonomously and apply them to the resolution of problems with the optimal software application.

While this course's evaluation is such that it does not specifically evaluate capabilities that directly contribute to the 2030 Agenda's achievement, the knowledge and skills obtained in it are a basis for the knowledge and skills that will be later obtained throughout the degree, and that can have a more direct relationship with SDOs and the 2030 Agenda.

1.2. Context and importance of this course in the degree

Computer Studies is a basic course during the first year of the degree. This particular tempoal location facilitates that the students can apply the adquired knowledge in subsequent courses through the whole degree, most of them require more or less support from computer tools for problem solving.

1.3. Recommendations to take this course

This course introduces the student to problem solving through a computer, in which the tool (the computer) is used since the beginning, both in terms of general usage as well as in particular aspects oriented towards specific problem solving. To tackle this course, the student should be willing to develop problem solving abilities using a computer, through a constant practical work that cannot be replace by any other learning technique.

2. Learning goals

2.1. Competences

After passing the course, the student will be more competent for...

Solving problems and make decisions with initiative, creativity and critic reasoning.

Applying information and communication technologies to Engineering.

Adequately and effectively using computers, operating systems, development environments, data bases and computer programs with application in Engineering.

2.2. Learning goals

The student will need to prove that he/she:

1. Has the ability to recover information (browsers, search engines...).
2. Knows about basic computer inner workings, operating systems and databases, and is able to build simple programs on top of them.
3. Is capable of operating computer equipment effectively, considering its logical and physical properties.
4. Knows and skillfully uses available tools and software applications.
5. Correctly understands a problem and identifies the options for its resolution. Applies the adequate resolution and assesses the correctness of a solution.
6. Is able to specify, design and build simple computer systems.

2.3. Importance of learning goals

This course is the first contact with abilities that are fundamental for an engineer, that are put to test with real problems since the beginning. Computer Science not only tackles the exploitation of computation and computer technology but is also fundamental for gaining problem solving skills.

3. Assessment (1st and 2nd call)

3.1. Assessment tasks (description of tasks, marking system and assessment criteria)

The propose evaluation activities are:

1. **Practical laboratory work (15%).** The skill on problem solving and computer usage will be assessed, as well as the implemented solutions for each suggested task for each laboratory session, focusing on the quality of the procedure, the strategy for problem solving as well as the quality of the solution that implements that strategy.
2. **Practical project (15%).** It will assess the capability of identifying the needs for solving problems and the ability to choose among different alternatives considering also the reasoning process towards such solution.
3. **Exam (70%).** Several problems will be presented, similar in complexity to the ones suggested during the course. This activity will assess both the quality and effectiveness of the solution as well as the correctness of the methodology that leads to such solution. Severe semantic errors as well as lack of knowledge of basic coding rules might mean full penalization of the whole problem.

The student will pass the course with a minimum weighted mark of 5/10 and a mark over 4/10 on every activity. In the case that the student does not reach the minimum mark in one or more activities, the mark for the full course will be the minimum between the weighted mark and 4/10.

Activities 1 and 2 can be evaluated throughout the whole course or with a specific exam at the official exam date.

4. Methodology, learning tasks, syllabus and resources

4.1. Methodological overview

The designed learning process is based on:

1. Classic blackboard teaching.
2. Problem-solving.
3. Self-studying.
4. Practical work, developing theoretical concepts.
5. The development of proposed programs of increasing difficulty.

The course has a twofold orientation including theory and practice. Therefore the learning process involves multiple activities ranging from assistance to classical blackboard teaching classes to practical lab work to student self-studying.

4.2. Learning tasks

This course's learning process involves the following activities:

- Classical blackboard classes: teacher analyzes and explores the course's concepts, illustrating them with examples.

- Problem-solving sessions: students will solve (in paper) problems following the course's concepts, guided by the teacher.
- Laboratory practical work: the student solves several programming problems that have to work in the computer, supervised by their teacher.
- Autonomous work: the student will solve bigger computing problems as described in specific materials provided to the student. The student has the possibility of requesting counseling from their teacher.
- Exam: evaluation activity with some problems that the student must solve.

4.3. Syllabus

The course will address the following topics:

- 0. Presentation
- 1. Introduction - computer architecture, operating systems, networks, machine language, assembler, compilers, introduction to programming.
- 2. Data types - internal representation, dominion, classification, integer, real, char, boolean.
- 3. Composition structures - sequential, conditional, iterative.
- 4. Behavior abstraction - procedures and functions.
- 5. Data abstraction - composed data types, arrays, records, strings.
- 6. Files - sequential, text.

4.4. Course planning and calendar

The course's calendar is defined by EINA in the official calendar for the whole study.

The expected workload is:

- Attending classes: 30 hours.
- Written problem solving: 15 hours.
- Supervised practical sessions: 12 hours.
- Individual practical work: 60 hours.
- Self-studying and problem solving: 30 hours.
- Exams, or other evaluation activities: 3 hours.

The student must be aware of the specific deadlines for each practical activity during the course, as well as exam dates and other evaluation activities.

4.5. Bibliography and recommended resources

Link:
http://biblos.unizar.es/br/br_citas.php?codigo=30007&year=2019