Algorithms and data structures

Academic Year: (2020 / 2021) **Review date: 04/01/2021 19:17:17**

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: SEGURA BEDMAR, ISABEL

Type: Compulsory ECTS Credits : 6.0

Year : 1 Semester : 2

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

- Programming
- Calculus

OBJECTIVES

Competences
CECRI1 (1 ECTS) Evidence Assessment: LAB CASE
CECRI6 (Algorithms) (1.5 ECTS) Evidence Assessment: TESTING, LAB CASE, GROUP AND INDIVIDUAL WORK
CECRI7 (TYPES AND DATA STRUCTURES) (2.5 ECTS) Evidence Assessment: TESTING, LAB CASE, GROUP AND INDIVIDUAL WORK
CGB4 (0.50 ECTS) Evidence Assessment: TESTING, LAB CASE
GGB5 (0.50 ECTS) Evidence Assessment: LAB CASE

- Capacity to analyze and synthesize (PO e).
- Capacity to organize and plan the work (PO d).
- Resolution of problems (PO e).
- Working as a team (PO d).
- Capacity to put in practice theory knowledge (PO e).

2. Specific Competences.

a. Cognitive (to know).

- General knowledge about algorithms (PO a).
- Understanding of basic data structures (PO k).
- Familiarity with advanced data structures (PO k).

b. Procedural/instrumental (to be able to do).

- To be able to design and analyze the algorithms complexity (PO a).
- To be able to understand and use different data structures (PO k).
- To be able to implement program solutions to specific problems using these tools (PO e).

c. Attitude (Being)

- Ability to solve problems through algorithms (PO e).
- Ability to clarify, simplify and efficiency of solving problems (PO e and k).
- Ability to question and conclude various solutions to any problem (PO e and k).

The learning outcomes are:

- Resolution by students of problems that must prove they have the ability to combine theory and practice. (PO a, e, k)
- Implementation and design of a data structures and their algorithms to develop a lab case (PO a, d, e, k)

DESCRIPTION OF CONTENTS: PROGRAMME

1. Introduction to Abstract Data Type

2. Linear Abstract Data Types
   b. Stacks
   c. Queues.
   d. Singly and doubly linked lists

3. Analysis of Algorithms.
   a. Empirical Analysis.
   b. Theoretical Analysis. Big-O functions. Best and worst cases.

4. Recursion I.

5. Trees
   a. General concepts
   b. Binary Trees
   c. Tree Traversals
   d. Search Binary Trees.
   e. How to balance a tree.

6. Graphs
   a. Implementations
   d. Graph trasversals.
   e. Dijkstra's algorithm (shortest path)

7. Recursion II: Divide and Conquer

LEARNING ACTIVITIES AND METHODOLOGY

1. Theory Lectures with the objective of acquire the cognitive specific competences (PO a and k).
2. Academic activities guided by the teacher:
2.1. With the teacher: to solve exercises devoted to analyze, design and implement cases with different level of complexity in collaboration with students (PO a and k). Some of the exercises will be carried out in computer laboratories (PO k).
2.2. Student work: Homework, individually or cooperatively, with exercises, implementation cases and basic readings from bibliography proposed by the teacher (PO k and e).
Moreover, these activities can be performed as:
a. Individual work consisting on developing solutions to the problems and exercises posed by the teacher.
b. Working cooperatively developing solutions to the problems proposed by the teacher (PO d).
3. Mid-term partial exams and final exam (PO a, e, k).
4. There will be a group tutorship for each small group to solve the queries and doubts of students.

ASSESSMENT SYSTEM

% end-of-term-examination/test:	40
% of continuous assessment (assigments, laboratory, practicals…):	60

In addition to serve as formative activity, the exercises and examinations serve to be used as evaluation measure.
Students should also try to solve a practical case study using the concepts learned during the course.
The evaluation includes the assessment of the guided academic activities and practical work according to the following weighting:

Se seguirá un sistema de evaluación continua que incluye las siguientes pruebas:

FIRST MID-TERM EXAM (15%) (Po a, e, k)
- Lessons: 1,2,3 and 4.
- Type: face to face (small group)
- Value: 15% (15 points over 100).
- Week: 6 (8-12 Marzo)

SECOND MID-TERM EXAM (15%) (Po a, e, k)
- Lessons: 3, 4 and 5.

% end-of-term-examination/test:	40
% of continuous assessment (assigments, laboratory, practicals…):	60

- Type: face to face (small group)
- Week: 12 (26-30 April)
- Value: 15% (15 points over 100).

LAB CASE (30%) (Po a, e, d, k).
- Type: lab case. It is individual work. It should be delivered via AulaGlobal.
- Three phrases:
  Phase 1: Linear Data Structures. (10 points). Deadline and Oral Exam: week 10 in seminar face-to-fase session (2 teachers).
  Phase 2: Trees. (10 points). Deadline and Oral Exam: week 14 (10-14 May) in seminar face-to-fase session (2 teachers).
  Phase 3: Graphs. (10 points). Deadline and Oral Exam: week 14 (10-14 May) in seminar face-to-fase session (2 teachers).

The lab case will be evaluated using unittests and the answers provided for each student in the oral exams for each phase. The oral exams are mandatory for all students. In these exams, the professor can ask questions about the solution. The student should be able to explain and discuss any detail of their solution. If the student does not attend to the oral exam, he/she will be evaluated as not presented.

FINAL EXAM (ORDINARY CALL). 40%. (Po a, e, d, k).
- Lessons: 1-7.
- Mandatory
- Type: exams with problems about data structures and their algorithms. It also contains questions about recursion and analysis of algorithms.
- Value: 40 points.
- Minimum mark: 16 points. The student should obtain at least 16 points in this exam in order to take into account the other marks of the continuous evaluation tests.

If the final exam takes place in person (it depends on covid-19), the solutions should be clear, clean and fully legible and understable. Those solutions that do not meet these criteria will not be evaluated.

The final mark for the course is obtained by summing all marks of the evaluation continuous system. To pass the course, it is necessary to obtain at least 50 points (over 100).

If a student decides not to follow the continuous assessment, he / she will be entitled to take a final exam (same date and place as the ordinary exam). The grade obtained in this exam is equivalent to 60% of the final grade.

Si el estudiante decide no seguir el sistema de evaluación continua, tendrá derecho a realizar un examen final (misma fecha y lugar que el examen ordinario). La nota final del estudiante será el 60% de la nota obtenida en dicho examen final.

In the extraordinary call, the final exam will be 100% of the grade. The continuous evaluation may be applied if it is more beneficial for the student (partial tests and 40% of the final exam grade).

A solution (for example, the implementations of data structures and their algorithms) will be considered correct only if it meets the following criteria:
- It satisfies the specifications described in the problem statement.
- Its implementation is correct. That is, in addition to not containing syntax errors, for each possible input it receives, it produces the correct output. The objective of unittests is that students can easily verify the correctness of their algorithms.
- The solution must be as efficient as possible, both in terms of temporal and spatial complexity.
- In the case of face-to-face exams that will be written by hand, they must be clean, legible and easy to understand. A solution that does not meet these criteria will not be evaluated.

% end-of-term-examination/test:	40
% of continuous assessment (assigments, laboratory, practicals…):	60

BASIC BIBLIOGRAPHY

- Karumanchi, N Data Structure and Algorithmic Thinking with Python: Data Structure and Algorithmic Puzzles. , CareerMonk Publications, 2015

- Michael T. Goodrich and Roberto Tamassia Data Structures and Algorithms in Python, , John Wiley & Sons, 2013

ADDITIONAL BIBLIOGRAPHY

- Isabel Segura Bedmar, Harith AlJumaily, Julian Moreno Schneider, Juan Perea & Nathan D. Ryan Algorithms and Data Structures, OCW-UC3M: http://ocw.uc3m.es/ingenieria-informatica/algorithms-and-data-structures, 2011

BASIC ELECTRONIC RESOURCES

- Isabel Segura Bedmar, Harith AlJumaily, Julian Moreno Schneider, Juan Perea & Nathan D. Ryan . ALGORITHMS AND DATA STRUCTURES: <a href="http://ocw.uc3m.es/ingenieria-informatica/algorithms-and-data-structures" target="_blank">http://ocw.uc3m.es/ingenieria-informatica/algorithms-and-data-structures</a>