

Academic Year: (2023 / 2024)

Review date: 09-02-2024

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: GARCIA SANCHEZ, JOSE DANIEL

Type: Compulsory ECTS Credits : 6.0

Year : 3 Semester : 1

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

- + Programming (Year 1 / Term 1)
- + Computer structure (Year 2 / Term 1)
- + Operating Systems (Year 2 / Term 2)

SKILLS AND LEARNING OUTCOMES

CB1. Students have demonstrated possession and understanding of knowledge in an area of study that builds on the foundation of general secondary education, and is usually at a level that, while relying on advanced textbooks, also includes some aspects that involve knowledge from the cutting edge of their field of study.

CB2. Students are able to apply their knowledge to their work or vocation in a professional manner and possess the competences usually demonstrated through the development and defence of arguments and problem solving within their field of study.

CB3. Students have the ability to gather and interpret relevant data (usually within their field of study) in order to make judgements which include reflection on relevant social, scientific or ethical issues.

CB4. Students should be able to communicate information, ideas, problems and solutions to both specialist and non-specialist audiences.

CB5. Students will have developed the learning skills necessary to undertake further study with a high degree of autonomy.

CG1. Students are able to demonstrate knowledge and understanding of concepts in mathematics, statistics and computation and to apply them to solve problems in science and engineering with an ability for analysis and synthesis.

CG3. Students can solve computationally with the help of the most advanced computing tools mathematical models coming from applications in science, engineering, economy and other social sciences.

CG4. Students are able to show that they can analyze and interpret, with help of computer science, the solutions obtained from problems associated to real world mathematical models, discriminating the most relevant behaviours for each application.

CG6. Students can search and use bibliographic resources, in physical or digital support, as they are needed to state and solve mathematically and computationally applied problems arising in new or unknown environments or with insufficient information.

CE10. Students have shown that they know and understand the algorithmic procedures to design and build programs that solve mathematical problems paying special attention to performance.

CE13. Students have shown that they understand how computers work, and the impact of their structure and operation on programs performance as well as their physical limitations.

CE14. Students have shown that they know the theory of grammars, languages and automatas and they can apply it to programming languages and domain specific languages analyzers as well as that they understand the translation process for high-level languages and most common optimizations.

CE19. Students have shown that they know multiprocessor architectures as well as multi-core processors and computing accelerators and that they can use such knowledge to improve performance of mathematical algorithms with special attention to parallelization of those algorithms.

RA1. Students must have acquired advanced cutting-edge knowledge and demonstrated indepth understanding of the theoretical and practical aspects of working methodology in the area of applied mathematics and computing.

RA2. Through sustained and well prepared argument and procedures, students will be able to apply their knowledge, their understanding and the capabilities to resolve problems in complex specialized professional and work areas requiring the use of creative and innovative ideas.

RA3. Students must have the capacity to gather and interpret data and information on which they base their conclusions, including where relevant and necessary, reflections on matters of a social, scientific,

and ethical nature in their field of study.

RA4. Students must be able to perform in complex situations that require developing novel solutions in the academic as well as in the professional realm, within their field of study.

RA5. Students must know how to communicate with all types of audiences (specialized or not) their knowledge, methodology, ideas, problems and solutions in the area of their field of study in a clear and precise way.

RA6. Students must be capable of identifying their own education and training needs in their field of study and the work or professional environment and organize their own learning with a high degree of autonomy in all types of contexts (structured or not).

OBJECTIVES

The main goal of this course is that the student knows the basic concepts around the computer architecture and the impact that they have in the performance of applications and computer systems.

To achieve this goal:

- + Advanced aspects of computer architecture are covered in detail such as optimizations in memory hierarchy, instruction level parallelism, or multiprocessors design.
- + Basic concept in concurrent programming and its relationship with the computer architecture are enforced.
- + Parallel programming concepts are introduced.
- + A constant concern about application performance and energy consumption is encouraged.

DESCRIPTION OF CONTENTS: PROGRAMME

1. Fundamentals of computer design.

1.1. Introduction.

1.2. Historical perspective.

1.3. Classification of Computers.

1.4. Parallelism.

1.5. Computer Architecture.

2. Performance Evaluation of computer systems.

2.1. Classification of architectures and evaluation.

2.2. Computer Systems Reliability.

3. Memory hierarchy.

3.1. Cache memory optimizations.

3.2. Advanced cache optimizations.

4. Instruction Level Parallelism.

4.1. Introduction to Instruction Level Parallelism.

4.2. Exploitation of Instruction Level Parallelism.

5. Multiprocessors.

5.1. Symmetric shared memory architectures.

5.2. Distributed shared memory.

5.3. Synchronization in shared memory.

5.4. Memory consistency models.

6. Models of parallel and concurrent programming.

6.1. Introduction to parallel programming: OpenMP.

6.2. Concurrent programming model: ISO C ++ Threads .

6.3. Memory consistency models in C ++.

LEARNING ACTIVITIES AND METHODOLOGY

** TEACHING ACTIVITIES **

THEORETICAL-PRACTICAL CLASSES [42 hours with 100% attendance, 1.52 ECTS].

In them the knowledge that students must acquire will be presented. They will receive the class notes and will have basic reference texts to facilitate the follow-up of the classes and the development of the subsequent work. Exercises will be solved by the student which will serve as a self-assessment and to acquire the necessary skills. Problem solving classes, in which the problems proposed to the students are developed and discussed.

WORKSHOPS AND / OR LABORATORY PRACTICES [10 hours with 100% attendance, 0.40 ECTS].

TUTORING [28 hours with 25% attendance, 1.12 ECTS].

Individualized assistance (individual tutorials) or in groups (collective tutorials) to students by the teacher.

STUDENT INDIVIDUAL OR GROUP WORK [70 hours with 0% attendance, 2.8 ECTS].

FINAL EXAM [4 hours 100% attendance, 0.16 ECTS]

The knowledge, skills and abilities acquired throughout the course will be assessed globally.

**** METHODOLOGIES ****

Seminars and lectures supported by computer and audiovisual aids.

Individual and group or cooperative work with the option of oral or written presentation.

Individual and group tutorials to resolve doubts and queries about the subject.

Internships and directed laboratory activities.

ASSESSMENT SYSTEM

Evaluations during the course: 24%

Individual or group labs: 36%

Final exam: 40%

ORDINARY CALL

Minimum mark in the exam: 3.5 points out of 10.

Minimum mark in each lab: 2 points out of 10.

Minimum mark in projects: 2 points out of 10.

The final grade will be increased by 1 point to those students who take all the continuous assessment tests, obtain more than 7 in the continuous assessment and at least 6 points in the final exam.

For those students who have not completed the continuous assessment process, or have not obtained the minimum marks in labs, the final exam will be worth 60% of the total grade for the course. Therefore, in order to pass, the student must obtain a grade higher than 8.33 out of 10 in this exam.

EXTRAORDINARY CALL

Students who do not pass the subject in the ordinary call will take a final exam in this call. The evaluation in the extraordinary call will be carried out as follows:

- a) If the student completed the continuous evaluation process (see previous section), the final exam of this call will have a weight of 50% and the final grade will take into account the other 40% obtained in the continuous evaluation process, always that a grade higher than 3.5 points has been obtained in the exam.
- b) If the student did not complete the continuous assessment process, the exam in this call will have a value of 100% of the final grade for the course.
- c) Although the student has followed the continuous evaluation process, the exam of this call will have a value of 100% if this criterion is more favorable.

% end-of-term-examination:	40
% of continuous assessment (assignments, laboratory, practicals...):	60

BASIC BIBLIOGRAPHY

- Hennessy, JL y Patterson, DA. Computer Architecture: A Quantitative Approach. 6th Edition., Morgan Kaufmann,, 2017

ADDITIONAL BIBLIOGRAPHY

- David A. Patterson, John Hennessy Computer Organization and Design MIPS Edition: The Hardware/Software Interface, Morgan Kaufmann, 2020

- Timothy G. Mattson, Yun (Helen) He and Alice E. Koniges The OpenMP Common Core: Making OpenMP Simple Again, MIT Press, 2019

- Williams, A. C++ Concurrency in Action. Practical Multithreading. 2nd Edition, Manning., 2018