Programming Techniques

Academic Year: (2023 / 2024) **Review date: 09-02-2024**

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: GRANADOS FONTECHA, ANA

Type: Basic Core ECTS Credits : 6.0

Year : 1 Semester : 2

Branch of knowledge: Engineering and Architecture

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

Differential Calculus (Course 1 - Semester 1)
Programming (Course 1 - Semester 1)

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.
CE11. Students have shown that they know the concepts of imperative, generic, object oriented and functional programming and distinguish interpreted, virtual machine based and native programming languages as well as the impact that they have on performance of algorithms and applications.

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 communication 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.

DESCRIPTION OF CONTENTS: PROGRAMME

1. Basics of C++ Language
2. Compound Data Type
3. Functions
4. Errors handling and Defensive programming
5. Object-Oriented Programming
6. Input/Output Streams
7. Dynamic Memory Management
8. Generic Programming
9. Containers, Iterators, and Algorithms

LEARNING ACTIVITIES AND METHODOLOGY

THEORETICAL-PRACTICAL CLASSES. [44 hours with 100% classroom instruction, 1.67 ECTS]
Knowledge and concepts students must acquire. Student receive course notes and will have basic reference texts to facilitate following the classes and carrying out follow-up work. Students partake in exercises to resolve practical problems and participate in workshops and evaluation tests, all geared towards acquiring the necessary capabilities.

TUTORING SESSIONS. [4 hours of tutoring with 100% on-site attendance, 0.15 ECTS]
Individualized attendance (individual tutoring) or in-group (group tutoring) for students with a teacher.

STUDENT INDIVIDUAL WORK OR GROUP WORK [98 hours with 0 % on-site, 3.72 ECTS]

WORKSHOPS AND LABORATORY SESSIONS [8 hours with 100% on site, 0.3 ECTS]

FINAL EXAM. [4 hours with 100% on site, 0.15 ECTS]
Global assessment of knowledge, skills and capacities acquired throughout the course.

METHODOLOGIES

THEORY CLASS. Classroom presentations by the teacher with IT and audiovisual support in which the subject's main concepts are developed, while providing material and bibliography to complement student learning.

PRACTICAL CLASS. Resolution of practical cases and problem, posed by the teacher, and carried out individually or in a group.

TUTORING SESSIONS. Individualized attendance (individual tutoring sessions) or in-group (group tutoring sessions) for students with a teacher as tutor.

LABORATORY PRACTICAL SESSIONS. Applied/experimental learning/teaching in workshops and laboratories under the tutor's supervision.

ASSESSMENT SYSTEM

SE1 - FINAL EXAM. [50 %]
Global assessment of knowledge, skills and capacities acquired throughout the course.

SE2 - CONTINUOUS EVALUATION. [50 %]
Projects, class presentations, debates, exercises, internships and workshops throughout the course.

The CONTINUOUS EVALUATION includes the assessment of the guided academic activities and practical work according to the following weighting:
1. Two Lab Tests: [20%]
2. Midterm exam: [30%].
3. FINAL EXAM: [50%]. Minimum Mark 4/10.

The final grade of the course is obtained by adding the grade of the different partial tests. In order to pass the course, it is necessary to obtain a final grade equal or higher than 5/10.

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

In the extraordinary call, the final exam will be 100% of the grade. However, the continuous evaluation can be applied if it is more beneficial for the student (the grades of the continuous evaluation tests [50%] + the grade of the extraordinary exam [50%]). To apply this option, it is necessary to obtain the minimum grade 4/10 in this exam.

% end-of-term-examination:	50
% of continuous assessment (assigments, laboratory, practicals…):	50

BASIC BIBLIOGRAPHY

- Bjarne Stroustrup The C++ Programming Language, Addison-Wesley, 2013
- Bjarne Stroustrup Programming: Principles and Practice Using C++, Second Edition, Addison-Wesley Professional, 2014
- Josuttis, Nicolai M. The C++ standard library: a tutorial and reference, 2nd Edition, Addison-Wesley, 2012
- Stanley, B., and Lajoie L. C++ PRIMER, Addison-Wesley, 2019
- Stroustrup, Bjarne A Tour of C++. 2nd Edition, Addison-Wesley Professional, 2018