



Course guide

200011 - INF - Computer Science

Last modified: 11/04/2024

Unit in charge: School of Mathematics and Statistics
Teaching unit: 723 - CS - Department of Computer Science.

Degree: BACHELOR'S DEGREE IN MATHEMATICS (Syllabus 2009). (Compulsory subject).

Academic year: 2024 **ECTS Credits:** 7.5 **Languages:** Catalan, Spanish

LECTURER

Coordinating lecturer: MARIA JOSE BLESA AGUILERA

Others: Primer quadrimestre:
MARIA JOSE BLESA AGUILERA - M-A1, M-A3, M-B2, M-B3
AMALIA DUCH BROWN - M-A1, M-B2
EMMA ROLLÓN RICO - M-A3, M-B3
SALVADOR ROURA FERRET - M-A1, M-A3, M-B2, M-B3

PRIOR SKILLS

Capability for abstract reasoning.

REQUIREMENTS

Knowledge of basic informatics tools at user level.

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Specific:

1. CE-2. Solve problems in Mathematics, through basic calculation skills, taking in account tools availability and the constraints of time and resources.
3. CE-4. Have the ability to use computational tools as an aid to mathematical processes.

Generical:

4. CB-1. Demonstrate knowledge and understanding in Mathematics that is founded upon and extends that typically associated with Bachelor's level, and that provides a basis for originality in developing and applying ideas, often within a research context.
5. CB-2. Know how to apply their mathematical knowledge and understanding, and problem solving abilities in new or unfamiliar environments within broader or multidisciplinary contexts related to Mathematics.
6. CB-3. Have the ability to integrate knowledge and handle complexity, and formulate judgements with incomplete or limited information, but that include reflecting on social and ethical responsibilities linked to the application of their knowledge and judgements.
7. CG-1. Show knowledge and proficiency in the use of mathematical language.
9. CG-3. Have the ability to define new mathematical objects in terms of others already know and ability to use these objects in different contexts.
10. CG-4. Translate into mathematical terms problems stated in non-mathematical language, and take advantage of this translation to solve them.
12. CG-6 Detect deficiencies in their own knowledge and pass them through critical reflection and choice of the best action to extend this knowledge.

Transversal:

2. SELF-DIRECTED LEARNING. Detecting gaps in one's knowledge and overcoming them through critical self-appraisal. Choosing the best path for broadening one's knowledge.

TEACHING METHODOLOGY

The theory classes present the basic theoretical basis necessary for building programs.

Problems sessions are designed for solving exercises with pencil and paper, to consolidate the acquired knowledge and to design algorithms for solving the statements posed. They are intended as a series of participatory sessions in which the student engages with ideas and present their solutions. Prior preparation by the student is required.

In the lab sessions, the student performs individually, with the help of teachers, programming exercises that demonstrate the use of the concepts taught in theory classes.

Over the course we introduce the theoretical components that must be assimilated by students. To this end, we believe that the most convenient method is the resolution of problems that require the tool or concept introduced. It is therefore essential student personnel work in the design and implementation of programs. This effort will be supported by self-learning tools.

As a complement self-learning tools will be provided so that students can consolidate their knowledge of programming during the hours of study outside the classroom. Specifically, students will have access to a version adapted to the contents of the subject of a self-learning tool for programming, the "Jutge" developed within the Department of Computer Science by a team of teachers led by Jordi Petit and Salvador Roura.

The subject will be taught preferably and mostly in Catalan, although there could be some groups in Spanish depending on the assigned teachers.

LEARNING OBJECTIVES OF THE SUBJECT

The overall objective of the course is that the student be able to write fluently and legibly correct programs to solve problems of medium difficulty, based on processing sequences, and basic difficulty in other areas, in particular problems with mathematical formulation. Another aim is to familiarize students with a computing environment and a current programming language, in this case C ++. Students must learn, first, to design and implement algorithms and, second, to use other tools such as editors and compilers.

Specific objectives:

- Making students feel comfortable and reliable in the design of programs written in an imperative language.
- Learn the basic algorithms with elementary and structured data (prime numbers, gcd, traversals, searching, sorting, matrices ...).
- To apply the inductive method for solving complex problems.
- To use editing, compilation and execution tools to code and run programs.

STUDY LOAD

Type	Hours	Percentage
Self study	105,0	56.00
Hours large group	30,0	16.00
Hours small group	45,0	24.00
Guided activities	7,5	4.00

Total learning time: 187.5 h



CONTENTS

1. The structure of a computer. Procedures and instructions.

Description:

Processes and instructions. Hardware and software. Basic structure of a computer. Computing environment. Programming languages. Compilers and interpreters. Programming and troubleshooting. Programs and algorithms. The software life cycle.

Basic orders in Linux. Text editors.

Full-or-part-time: 15h

Theory classes: 2h 30m

Practical classes: 3h

Self study : 9h 30m

2. Variables and basic instructions.

Description:

Data Types: domain and operations. Kinds of expressions. Assignment. Alternative composition. Iterative composition. Basic algorithms.

Completion and correction.

Basic syntax of the C + +. Writing, compiling and running a program in C + +.

Full-or-part-time: 31h 30m

Theory classes: 5h 30m

Practical classes: 6h

Self study : 20h

3. Treatment sequences.

Description:

The concept of sequence. Traversing and searching. Examples. Traversing and searching schemes.

Full-or-part-time: 41h

Theory classes: 7h

Practical classes: 10h

Self study : 24h

4. Actions and functions.

Description:

Parameter concept. Implementation of mechanisms for parameter passing. Actions and functions. Examples.

Introduction to recursion.

Methods and functions in C + +. Side effects.

Full-or-part-time: 29h 30m

Theory classes: 5h 30m

Practical classes: 5h

Self study : 19h

5. Data not elementary.

Description:

Arrays. Matrix representation. Algorithms for matrix algebra (addition, symmetric matrix, matrix transpose, matrix multiplication). Sorting algorithms (insertion, selection, bubble, radix).

Down design. Efficiency.

The vector class. C++ syntax.

Full-or-part-time: 41h

Theory classes: 7h

Practical classes: 10h

Self study : 24h

6. Tuples.

Description:

Non-homogeneous data. Basic notions of objects. Examples of use.

Full-or-part-time: 28h

Theory classes: 5h

Practical classes: 5h

Self study : 18h

7. Fundamental algorithms.

Description:

Study of well-known fundamental algorithms in Computer Science.

Classification of problems in relation to the existence of algorithmic solutions and their efficiency.

Full-or-part-time: 11h 30m

Theory classes: 3h 30m

Self study : 8h

GRADING SYSTEM

The assessment takes into account the following components:

- Knowledge and use of algorithms and techniques introduced in the course
- Algorithmic problem-resolution.
- Ability to program in C++ and Python simple programs.
- Ability to problem-solving mid-level programming.

There will be a programming partial test (PAR) and a final test (FIN), to take place in the laboratory rooms.

The final grade is calculated according to the formula:

$$\max\{ (0.4 \text{ PAR} + 0.6 \text{ FIN}), \text{FIN}\}$$

An extra exam will take place on July for students that failed during the regular semester.



EXAMINATION RULES.

The "Judge" will be used in conducting laboratory tests, partial and final, providing the same software development environment, to aid them during the tests. This tool will also support the development of the programming project.
In none of the tests it is allowed the use of books or notes.

BIBLIOGRAPHY

Basic:

- Franch Gutiérrez, Xavier [et al.]. Informàtica bàsica [on line]. 3a ed. Barcelona: Edicions UPC, 2002 [Consultation: 21/05/2020]. Available on: <http://hdl.handle.net/2099.3/36244>. ISBN 8483016605.
- Savitch, Walter J. Problem solving with C++. 8th ed. Boston: Addison Wesley, 2012. ISBN 9780321412690.
- Beekman, George. Introducció a la informàtica. 6ª ed. Madrid: Pearson Educació, 2005. ISBN 8420543454.
- Xhafa, Fatos [et al.]. Programació en C++ para ingenieros. Madrid: Thomson, 2006. ISBN 8497324854.
- Oualline, Steve. Practical C++ programming [on line]. O'Reilly, 2003 [Consultation: 15/06/2023]. Available on: <https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pg-origsite=primo&docID=540759>. ISBN 9780596523145.
- Lutz, Mark. Learning Python [on line]. O'Reilly, 2009 [Consultation: 26/06/2023]. Available on: <https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pg-origsite=primo&docID=1224732>. ISBN 9780596158064.

Complementary:

- Cohen, Edward. Programming in the 1990s : an introduction to the calculation of programs. Study ed. New York: Springer-Verlag, 1990. ISBN 0387973826.
- Vancells, Joan; López i Ruestes, Enric. Programació: introducció a l'algorísmica. Barcelona: Eumo, 1992. ISBN 8476025610.
- Sipser, Michael. Introduction to the theory of computation [on line]. 3rd ed. Boston: Thomson Course Technology, 2012 [Consultation: 26/06/2023]. Available on: <https://fuuu.be/polytech/INFOF408/Introduction-To-The-Theory-Of-Computation-Michael-Sipser.pdf>. ISBN 0619217642.
- Hromkovic, Juraj. Algorithmic adventures : from knowledge to magic [on line]. Berlin: Springer, 2009 [Consultation: 26/06/2023]. Available on: <https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pg-origsite=primo&docID=450685>. ISBN 9783540859857.