COURSE DATA

DATA SUBJECT

Code: 34686
Name: Advanced programming
Cycle: Undergraduate Studies
ECTS Credits: 6
Academic year: 2025-26

STUDY (S)

Degree	Center	Acad. year	Period
1400 - Degree in Computer Engineering	Escola Tècnica Superior d'Enginyeria	4	First quarter
1407 - Degree in Multimedia Engineering	Escola Tècnica Superior d'Enginyeria	4	First quarter

SUBJECT-MATTER

Degree	Subject-matter	Character
1400 - Degree in Computer Engineering	Optional subject	ELECTIVES
1407 - Degree in Multimedia Engineering	Optatividad	ELECTIVES

COORDINATION

BARBER MIRALLES FERNANDO

SUMMARY

The course "Advanced Programming" is a subject of the fourth year of the Degree of Computer Engineering.

This subject is thought to teach programming techniques especially suitable to improve program efficiency, including for example templates and metaprogramming.

The subject is focused on C++. The latest standards of C++ will be studied (C++11, C++14, ...) and C++ will be the programming language for all the works.

These techniques are very interesting in game programming and other systems, where efficiency is a must. They also allow taking full advantage of C++.

/strong>

PREVIOUS KNOWLEDGE

RELATIONSHIP TO OTHER SUBJECTS OF THE SAME DEGREE

There are no specified enrollment restrictions with other subjects of the curriculum.

OTHER REQUIREMENTS

It is highly desirable that students have taken the courses "Informatics", "Programming" and "Data Structures and Algorithms". It is necessary to have an intermediate level of programming in C++. Also, it is desirable to have some basic knowledge of processor architecture.

COMPETENCES / LEARNING OUTCOMES

1400 - Degree in Computer Engineering

C2 - Ability to acquire, obtain, formalise and represent human knowledge in a computable form for solving problems through a computer system in any field, particularly in those related to aspects of computing, perception and action in intelligent environments.

1405 -

Program correctly in the different specific languages of multimedia systems taking into account time and cost restrictions.

DESCRIPTION OF CONTENTS

1. Introduction to efficient programming

-General strategies. Profiling.
-Sorting with no comparisons. Countsort. Radixsort.
-Amortized analysis of algorithms. Expansible vector

2. Standard C++11

- Modifications to the core language.
- Modifications to the standard library.

3. Efficiency improvement in C++

-Common optimizations in C++.
-Dynamic Memory in C++. New and delete modifications.

4. Use of templates

-Template functions.
-Template classes.
-Variadic Templates.

5. Advanced use of templates

-Polymorphism and templates.
-Traits templates and Policy classes.
- Metaprogramming.
- Expression templates: Efficient numeric arrays.
- Efficient numeric arrays in other languages: Python.

6. Efficiency improvement in recursive programs

-Function embedding.
-Unfold/Fold transformation. Recursive-Iterative transformation.
-Dynamic programming. Floyd algorithm.

WORKLOAD

PRESENCIAL ACTIVITIES

Activity	Hours
Theory	30,00
Laboratory	20,00
Classroom practices	10,00
Total hours	**60,00**

NON PRESENCIAL ACTIVITIES

Activity	Hours
Attendance at other activities	0,00
Individual or group project	26,00
Independent study and work	10,00
Preparation of lessons	39,00
Preparation for assessment activities	15,00
Resolution of case studies	0,00
Total hours	**90,00**

TEACHING METHODOLOGY

The course themes will be developed at the classroom based activities providing general as well as

integrated view, analysing in details and of major complexity the key aspects and encouraging students' participation at every point. These activities are complemented by practical activities in order to apply basic concepts and expand them with the knowledge and experience which is gained during the performance of the proposed works. They include the following classroom activities:

- Lessons of problems and questions in classroom

- Sessions of discussions, solving of problems and exercises previously worked by students

- Laboratory practice

- Classroom evaluation through individual questionnaires at the presence of the professorship

Besides the classroom activities the students are to work at individual tasks (out of the classroom) like: monograph works, guided bibliographic search, issues and problems, as well as preparation for lessons and exams. These tasks mainly will be carried out individually in order to upgrade skills of independent work, but in addition will be included some projects which will require the small group (2-4 students) participation with the purpose to build up the capacity for integration into work groups.

The e-learning platform (Aula Virtual) is used in the University of Valencia as a form of communication with students. Students have access through it to the teaching materials used at lessons as well as to problems and exercises to be resolved.

EVALUATION

The evaluation of the course is carried out according to the following scheme:

- Continuous evaluation (*N_Continua*) based on the degree of participation and involvement in the teaching-learning process, taking into account regular assistance at the planned classroom activities, resolution of issues and problems and of works to be delivered.
- Individual objective test (*N_Examenes*) consisting of several tests or knowledge tests which include both theoretical and practical issues as well as problems. Some of the tests may be substituted with works.
- Evaluation of practical activities (*N_Practicas*) taking into account the achievement of objectives in the lab sessions, resolving problems and projects elaboration.

The final course grade will be calculated according to the following formula:

Final Grade = 20% N_Continua + 50% N_Examenes + 30% N_Practicas

It is required to obtain a minimum grade of 4.5 out of 10 in N_Examenes and N_Practicas to pass the course.

The grade of N_Continua is not recoverable. The grade is maintained in 2º call.

In any case, the evaluation of this subject will be done in compliance with the University Regulations in this regard, approved by the Governing Council on 30th May 2017 (ACGUV 108/2017)

Copying or plagiarism of any activity that is part of the evaluation will result in the impossibility of passing the course, and the student will then be subject to the appropriate disciplinary procedures indicated in the ACTION PROTOCOL FOR FRAUDULENT PRACTICES AT THE UNIVERSITY OF VALENCIA (ACGUV 123/2020).

REFERENCES

- B. Stroustrup, The C++ Programming Language, 4ª Ed., Addison-Wesley, 2013.
- T.H. Cormen et al, Introduction to Algorithms, 4ª Ed., The MIT Press, 2022.
- D. Vandevoorde, N.M. Josuttis, C++ Templates: The Complete Guide, 2ª Ed. Addison-Wesley, 2017.
- N.M. Josuttis, "The C++ Standard Library: A Tutorial and Reference", 2º Ed. Addison-Wesley. 2012.
- P. Deitel, H. Deitel, "C++ How to Program", 9ª Ed, Prentice Hall. 2013.
- S. Meyers, "Effective C++: 55 Specific Ways to Improve your Programs and Designs", 3ª Ed. Addison-Wesley. 2005.
- S. Meyers, "More Effective C++: 35 New Ways to Improve your Programs and Designs", Addison-Wesley. 1995.
- S. Meyers, "Effective STL: 50 Specific Ways to Improve your Use of the Standard Template Library", Addison-Wesley. 2001.
- A. Alexandrescu, "Modern C++ Design: Generic Programming and Design Patterns Applied", Addison-Wesley. 2001.

- M. Gregoire, "Professional C++", John Wiley & Sons. 2018.
- S. Meyers, "Effective Modern C++", Ed. OReilly Media, 2014.
- B. Stroustrup, A tour of C++, 3ª Ed, Addison-Wesley, 2022.