Course guide
804227 - PROG2VJ - Programming II

Last modified: 16/07/2025

Unit in charge: Image Processing and Multimedia Technology Centre
Teaching unit: 804 - CITM - Image Processing and Multimedia Technology Centre.

Degree: BACHELOR'S DEGREE IN VIDEO GAME DESIGN AND DEVELOPMENT (Syllabus 2014). (Compulsory subject).

Academic year: 2025 **ECTS Credits:** 6.0 **Languages:** Catalan, English

LECTURER

Coordinating lecturer: Mateo, Ramon

Others: Mateo, Ramon

TEACHING METHODOLOGY

Lectures are divided in 2h sessions. During lectures, the teacher presents the theoretical concepts and explains them by means of examples that are solved in class. Some time is also dedicated to the resolution of exercises by the students with the assistance of the teacher (solve the doubts that may appear).

An extensive use of campus virtual is mandatory, since the material of the course (slides, exercises, exams, etc.) and the communication between the students and the teacher are done through this system.

LEARNING OBJECTIVES OF THE SUBJECT

- To distinguish the concepts related to programming in the field of video game design and development: algorithm, type, value, variable, recursion, interpretation and compilation.

- To apply functions and data structures complying with certain specifications: stack, queue, linked list.

- To apply recursive solutions to simple problems.

- To implement simple objects with classes from a specification.

- To develop programs capable of: performing line-by-line input/output actions on text files, using control, repetition and selection structures, handling strings, vectors, registers and arrays, and using data streams.

STUDY LOAD

Type	Hours	Percentage
Hours medium group	16,0	10.67
Guided activities	20,0	13.33
Hours large group	24,0	16.00
Self study	90,0	60.00

Total learning time: 150 h

CONTENTS

1. Pointers and memory management

Description:
Description:
· Pointers
· Pointers and references as parameters
· Management of memory dynamically: allocation and deallocation of resources

Full-or-part-time: 20h
Theory classes: 4h
Guided activities: 4h
Self study : 12h

2. Object oriented programming

Description:
Description:
· Classes and modular design
· Overloading of operators
· Inheritance
· Polymorfism
· Friendship and templates

Full-or-part-time: 45h
Theory classes: 8h
Guided activities: 10h
Self study : 27h

3. Data structures

Description:
Description:
· Strings
· Stacks and queues
· Lists
· Dynamic arrays
· Trees

Full-or-part-time: 40h
Theory classes: 8h
Guided activities: 8h
Self study : 24h

4. Recursion and Fractals

Description:
Structure of recursive algorithms
Recursion in numeric series
The Hanoi Towers
Fractals: The Cantor set and the Sierpinski triangle

Full-or-part-time: 20h
Theory classes: 4h
Guided activities: 4h
Self study : 12h

5. Sorting algorithms

Description:
Description:
· Bubblesort
· Quicksort
· Binary Tree
· Heap Sort
· Merge Sort

Full-or-part-time: 25h
Theory classes: 6h
Guided activities: 4h
Self study : 15h

ACTIVITIES

Activities about pointers and memory

Description:
Two sets of questions and small exercices will allow to practice with the concepts of pointers and memory management.
The first set of exercices corresponds to the implementation and management of basic variables and pointers to them, modifying both their address and the content of the memory where they point to.
The second set of exercices includes structs, defyning variables that will be either instances of those struct or pointers to them, allowing to start working with the operators for accessing to the members of the struct, that latter on will be used for the objects and their members.

Specific objectives:
To practice with the theoretical contents expossed and get practice with the implementation of programmes in C++

Material:
Statement of the exercices and MS Visual C++

Delivery:
The output of this activity is a set of C++ files to be uploaded to the delivery section in the campus and eventually presented in front of the group in the classroom. These deliveries and presentations will be used to measure the 2% of the mark corresponding to the student's participation.

Full-or-part-time: 8h
Self study: 4h
Practical classes: 4h

Activities of Foundations of Object Oriented Programming

Description:
Set of programming exercices to practice with the concepts of encapsulation, inheritance and polymorphism. Among others they will target the representation and manipulation of bubbles, capsules, vehicles, Pokemons and vectors.

Specific objectives:
To practice with the theoretical contents expossed and get practice with the implementation of programmes in C++

Material:
Statement of the exercices and MS Visual C++

Delivery:
The output of this activity is a set of C++ files to be uploaded to the delivery section in the campus and eventually presented in front of the group in the classroom. These deliveries and presentations will be used to measure the 2% of the mark corresponding to the student's participation.

Full-or-part-time: 18h
Self study: 10h
Practical classes: 8h

Activities of Data Structures

Description:
Questionnaire (quiz game) in the classroom with questions to review the concepts explained.
Set of programming exercices to implement objects linked to basic structures and to practice by using them with some examples, namely: i) Stacks, ii) Queues, iii) Lists, iv) Dynamic Arrays, v) Matrix and vi) Trees.
These exercices will contribute to reinforce the training in object oriented programming.

Specific objectives:
To practice with the theoretical contents expossed and get practice with the implementation of programmes in C++
Be able to identify the apropiate data structures to solve different problems and situations

Material:
Statement of the exercices and MS Visual C++

Delivery:
The output of this activity is a set of C++ files to be uploaded to the delivery section in the campus and eventually presented in front of the group in the classroom. These deliveries and presentations will be used to measure the 2% of the mark corresponding to the student's participation.

Full-or-part-time: 18h
Self study: 8h
Practical classes: 10h

Activities for recursion and fractals

Description:
To Practice and solve some problems and games intrinsically resursive.
Set of programming exercices for the implementation of solutions to the problems and games used, including numerical series, Hanoi towers and fractals

Specific objectives:
Acquire the understanding of the recursion programming concept
Be able to implement recursive algorithms

Material:
Statement of the exercices and MS Visual C++

Delivery:
The output of this activity is a set of C++ files to be uploaded to the delivery section in the campus and eventually presented in front of the group in the classroom. These deliveries and presentations will be used to measure the 2% of the mark corresponding to the student's participation.

Full-or-part-time: 8h
Self study: 4h
Practical classes: 4h

Activities for sorting algorithms

Description:
Practice different sorting methods in mumerical series
Set of programming exercices to implement and use i) bubblesort, ii) heapsort y iii) quicksort
Include the data structures required in each sorting method

Specific objectives:
Identify and use the apropiate data structures for each sorting method
To know how to apply and understand the functioning of the recursive algorithms in the sorting methods that use them
Be able to use the sorting algorithms to sort different objects in different scenarios

Material:
Statement of the exercices and MS Visual C++

Delivery:
The output of this activity is a set of C++ files to be uploaded to the delivery section in the campus and eventually presented in front of the group in the classroom. These deliveries and presentations will be used to measure the 2% of the mark corresponding to the student's participation.

Full-or-part-time: 8h
Self study: 4h
Practical classes: 4h

GRADING SYSTEM

There are three types of tasks to measure the progress of the student:
- Four tests to evaluate specific topics of the course that are completed in class (PE)
- A midterm exam that is done in the midterms week specified in the academic calendar (ME)
- The final exam (FE)

The mark of the subject (M) is computed as follows:
NF = 0.15* EP + 0.4 * ET + 0.35 * EF + 0.1 * PART

where ME is the mark of the midterm exam, PE are the 4 practical exercises that will be done throughout the course (note that the mark each one corresponds to the 10% of the final mark of the subject) and FE is the mark of the final exam. Participation and learning attitude (PART) is also considered. This part will be evaluated according to the participation of the student in class (solving exercises, proposing solutions/alternatives), the interest shown in learning the different topics, etc.

The students who do not pass the course will have the possibility to do a reevaluation exam (percentage corresponding to the midterm and final exams is evaluated, and 5 will be the maximum mark of the subject).

Irregular actions that may lead to a significant variation of the grade of one or more students constitute a fraudulent performance of an evaluation act. This action entails the descriptive grade of failure and a numerical grade of 0 for the ordinary global evaluation of the course, without the right to re-evaluation.

If the lecturers have indications of the use of AI tools not allowed in the evaluation tests, they may summon the students concerned to an oral test or a meeting to verify the authorship.

EXAMINATION RULES.

Conducting different tests will be conducted exclusively through the mechanisms established on the dates and times indicated.

BIBLIOGRAPHY

Basic:
- Stroustrup, B. A tour of C++. Addison-Wesley, 2013. ISBN 9780321958310.
- Kernighan, Brian W; Ritchie, Dennis M. The C programming language. 2nd ed. New Jersey: Prentice Hall, 1988. ISBN 0131103628.
- Franch Gutiérrez, Xavier [et al.]. Fonaments de programació: problemes resolts en C++ [on line]. Barcelona: Edicions UPC, 2006 [Consultation: 10/12/2018]. Available on: http://hdl.handle.net/2099.3/36692. ISBN 9788483018828.

RESOURCES

Other resources:
C++ Reference: http://www.cplusplus.com/reference/ />C and C++ Programming: http://www.cprogramming.com/