

## Course guides

### 804227 - PROG2VJ - Programming II

**Last modified:** 02/09/2021

<b>Unit in charge:</b>	Image Processing and Multimedia Technology Centre	
<b>Teaching unit:</b>	804 - CITM - Image Processing and Multimedia Technology Centre.	
<b>Degree:</b>	BACHELOR'S DEGREE IN VIDEO GAME DESIGN AND DEVELOPMENT (Syllabus 2014). (Compulsory subject).	
<b>Academic year:</b> 2021	<b>ECTS Credits:</b> 6.0	<b>Languages:</b> Catalan, Spanish, English

#### LECTURER

---

<b>Coordinating lecturer:</b>	Löpfe, Lasse
<b>Others:</b>	Sangüesa, Ramon

#### DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

---

##### Specific:

3. Use programming languages, algorithmic patterns, data structures, visual programming tools, game engines and libraries for the development and prototyping of video games, in any genre and for any platform and mobile device.

##### Generical:

1. Interpret the basics of the use and programming of computers, operating systems, databases and in general, computer programs with applications for engineering.
2. Interpret and master the basics of discrete mathematics, logic, algorithmics and computational complexity, and their application to the automatic processing of information using computer systems and their application for solving engineering problems.

##### Transversal:

4. EFFICIENT ORAL AND WRITTEN COMMUNICATION. Communicating verbally and in writing about learning outcomes, thought-building and decision-making. Taking part in debates about issues related to the own field of specialization.
5. FOREIGN LANGUAGE: Achieving a level of spoken and written proficiency in a foreign language, preferably English, that meets the needs of the profession and the labour market.
7. EFFECTIVE USE OF INFORMATION RESOURCES: Managing the acquisition, structuring, analysis and display of data and information in the chosen area of specialisation and critically assessing the results obtained.

#### 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

- Consolidate the basic knowledge on pointers and memory management.
- Understand the basics on object oriented programming and be able to apply them adequately.
- Learn to use and implement the basic data structures: arrays, sequential data structures (stacks, queues and lists), trees and graphs.
- Learn the basis of recursion, searching and fractal generation.
- Learn, analyse and be able to implement and adapt the main sorting and search algorithms.

## STUDY LOAD

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

**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

#### Related competencies :

CGFB4VJ. Interpret the basics of the use and programming of computers, operating systems, databases and in general, computer programs with applications for engineering.

CEVJ 5. Use programming languages, algorithmic patterns, data structures, visual programming tools, game engines and libraries for the development and prototyping of video games, in any genre and for any platform and mobile device.

#### 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

### Related competencies :

CGFB4VJ. Interpret the basics of the use and programming of computers, operating systems, databases and in general, computer programs with applications for engineering.

CEVJ 5. Use programming languages, algorithmic patterns, data structures, visual programming tools, game engines and libraries for the development and prototyping of video games, in any genre and for any platform and mobile device.

**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

### Related competencies :

CGFB4VJ. Interpret the basics of the use and programming of computers, operating systems, databases and in general, computer programs with applications for engineering.

CEVJ 5. Use programming languages, algorithmic patterns, data structures, visual programming tools, game engines and libraries for the development and prototyping of video games, in any genre and for any platform and mobile device.

**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

**Related competencies :**

CGFB4VJ. Interpret the basics of the use and programming of computers, operating systems, databases and in general, computer programs with applications for engineering.  
CEVJ 5. Use programming languages, algorithmic patterns, data structures, visual programming tools, game engines and libraries for the development and prototyping of video games, in any genre and for any platform and mobile device.

**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

**Related competencies :**

CGFB4VJ. Interpret the basics of the use and programming of computers, operating systems, databases and in general, computer programs with applications for engineering.  
CEVJ 5. Use programming languages, algorithmic patterns, data structures, visual programming tools, game engines and libraries for the development and prototyping of video games, in any genre and for any platform and mobile device.

**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 exercises will allow to practice with the concepts of pointers and memory management. The first set of exercises 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 exercises includes structs, defining 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 exposed and get practice with the implementation of programmes in C++

**Material:**

Statement of the exercises 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.

**Related competencies :**

CGFB4VJ. Interpret the basics of the use and programming of computers, operating systems, databases and in general, computer programs with applications for engineering.

CEVJ 5. Use programming languages, algorithmic patterns, data structures, visual programming tools, game engines and libraries for the development and prototyping of video games, in any genre and for any platform and mobile device.

**Full-or-part-time:** 8h

Practical classes: 4h

Self study: 4h

### Activities of Foundations of Object Oriented Programming

**Description:**

Set of programming exercises 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 exposed and get practice with the implementation of programmes in C++

**Material:**

Statement of the exercises 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.

**Related competencies :**

CGFB4VJ. Interpret the basics of the use and programming of computers, operating systems, databases and in general, computer programs with applications for engineering.

CEVJ 5. Use programming languages, algorithmic patterns, data structures, visual programming tools, game engines and libraries for the development and prototyping of video games, in any genre and for any platform and mobile device.

**Full-or-part-time:** 18h

Practical classes: 8h

Self study: 10h



### Activities of Data Structures

**Description:**

Questionnaire (quiz game) in the classroom with questions to review the concepts explained.

Set of programming exercises 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 exercises will contribute to reinforce the training in object oriented programming.

**Specific objectives:**

To practice with the theoretical contents exposed and get practice with the implementation of programmes in C++

Be able to identify the appropriate data structures to solve different problems and situations

**Material:**

Statement of the exercises 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.

**Related competencies :**

CEVJ 5. Use programming languages, algorithmic patterns, data structures, visual programming tools, game engines and libraries for the development and prototyping of video games, in any genre and for any platform and mobile device.

**Full-or-part-time:** 18h

Practical classes: 10h

Self study: 8h

### Activities for recursion and fractals

**Description:**

To Practice and solve some problems and games intrinsically recursive.

Set of programming exercises 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 exercises 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.

**Related competencies :**

CEVJ 5. Use programming languages, algorithmic patterns, data structures, visual programming tools, game engines and libraries for the development and prototyping of video games, in any genre and for any platform and mobile device.

**Full-or-part-time:** 8h

Practical classes: 4h

Self study: 4h



### Activities for sorting algorithms

**Description:**

Practice different sorting methods in numerical series  
Set of programming exercises 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 appropriate 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 exercises 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

Practical classes: 4h

Self study: 4h

## GRADING SYSTEM

There are three types of tasks to measure the progress of the student:

- Four test (practical exercises) 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 (90% corresponding to the exams is evaluated and 5 will be the maximum mark of the subject).

## EXAMINATION RULES.

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

## BIBLIOGRAPHY

**Basic:**

- Kernighan, Brian W; Ritchie, Dennis M. The C programming language. 2nd ed. New Jersey: Prentice Hall, 1988. ISBN 0131103628.
- Stroustrup, B. A tour of C++. Addison-Wesley, 2013. ISBN 9780321958310.
- 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/>