

# COURSE INFORMATION

## PROGRAMMING FUNDAMENTALS

Code number: 606010104

Degree in Computer Engineering

Department: Information technologies

Academic Year: 2017-2018.

Course type: Foundational course. 1<sup>st</sup> year

First semester: 2 + 2 (theory, lab) hours a week

6 ECTS

Link to Spanish counterpart: <http://www.uhu.es/etsi/guia-de-assignatura/?codigo=606010104>

## TEACHING STAFF

Prof.: Manuel de la Villa

Department: Information Technologies.

Office: TU41 (Torreumbría Building, Campus La Rábida)

Phone: +34 959 217639

E-mail: manuel.villa@dti.uhu.es

Office hours:

First Semester:

(see updated info in <http://uhu.es/dti/files/tutorias.pdf>)

## PROGRAMME

### 1. DESCRIPTION

Fundamentals of programming is a first-year, first-semester course that provides an introduction to the fundamental principles in programming with particular focus on the concepts of algorithms and object oriented programming. This course is specially intended for students in the first year of a Computer Science degree.

Topics covered include: Programming languages. Object Oriented Programming Algorithms. Program Design Techniques.

- Introduction to Programming Languages
  - Definition and history of programming languages
  - Programming Paradigms
  - Compilers and Interpreters.
- Introduction to Object Oriented Programming Algorithms and Data Types.
  - Algorithms. Data Types, Operators and Expressions.
  - Control Structures.
  - Basic Types of Structured Data.
  - Methods and interfaces.
- Program Design Techniques.
  - Design Descending.
  - Modular Design.

### 2. PREREQUISITES

No prerequisites.

Students should study the subject with the support of suggested literature and attending regularly to tutorials. Due to the practical nature of the subject, it is recommended that students do most of the exercises included in problem sets.

### **3. LEARNING OUTCOMES**

Expressed as learning outcomes, the objectives are:

- To develop a disciplined approach to the specification, implementation, verification and documentation of programs.
- To appreciate the central role of abstraction in the task of programming
- To develop in the student the ability to solve problems using algorithm design techniques and apply it to program coding.
- To understand and use basic data structures, algorithms and general-purpose schemes.
- Learn a structured, general and extended high-level programming language that uses the object-oriented programming paradigm.
- Provide the theoretical and practical fundamentals for further studies in programming.

### **4. COMPETENCES**

SPECIFIC COMPETENCES: CB04, CB05

BASIC, GENERAL OR TRANSVERSAL COMPETENCES: CB1, CB5

### **5. TEACHING METHODOLOGY**

Teaching techniques used:

- Master classes.
- Supervised resolution of exercises
- Laboratory (computers) classes.
- Essay/tasks delivery.
- Evaluable exams.

### **6. CONTENTS**

THEORETICAL CLASSES:

SECTION 1: INTRODUCTION TO PROGRAMMING LANGUAGES

TOPIC 1: PROGRAMMING LANGUAGES.

1.1. Concept and history of programming languages.

1.2. Programming Paradigms.

1.3. Classification. Language of low, medium and high level.

1.4. Translators: Compilers and Interpreters. Executable code generation process. Compilation and linking.

1.5. Object-oriented programming.

SECTION 2: ALGORITHMS AND DATA TYPES

TOPIC 2: ALGORITHMS. DATA TYPES, OPERATORS AND EXPRESSIONS

2.1. Concept of algorithm. General structure.

2.2. Keywords, Identifiers, constants and comments. Variables and objects. 2.3. Data Types and Classes. Classification.

2.4. Assignment operators, arithmetic, relational and logical.

2.5. Expressions and order of precedence.

- 2.5.1. Rules of precedence and associativity.
- 2.5.2. Type conversions in expressions.
- 2.5.3. Coherence of types. Molding of types.
- 2.6. Basic Input / Output operations.

### TOPIC 3. CONTROL STRUCTURES

- 3.1. Sequential sentences.
- 3.2. Conditional sentences.
- 3.3. Iterative sentences.
- 3.4. Sentence macros.

### TOPIC 4. TYPES OF STRUCTURED DATA

- 4.1. Records and Records Structures. Hierarchical records.
- 4.2. Vectors and Matrices. Record Vectors
- 4.3. Classes and Objects.
- 4.4. Private and public statements. Scopes of visibility.

### SECTION 3: PROGRAM DESIGN TECHNIQUES

#### TOPIC 5: DESCENDING DESIGN.

- 5.1. Structured Programming.
- 5.2. Global declarations and local declarations. Scopes of visibility.
- 5.3. Functions and Methods: Interfaces.
- 5.4. Parametrization. Parameters by value and by reference.
- 5.5. Constructors and destroyers.
- 5.6. Passing complex data structures to functions and methods.
- 5.7. Overloading methods and operators.

#### TOPIC 6: MODULAR DESIGN

- 6.1. Modular Programming.
- 6.2. Concept of module.
- 6.3. Creation and use of modules
- 6.4. Import and export.
- 6.5. Definition and use of Modules.

#### LABORATORY CLASSES:

##### SECTION 2: ALGORITHM AND DATA TYPES.

Practices 1, 2 and 3. Introduction to C ++, use of classes, Use of strings and tables.

##### SECTION 3: PROGRAM DESIGN TECHNIQUES

Practice 4. Descending design.

## 7. BIBLIOGRAPHY

METODOLOGÍA DE LA PROGRAMACIÓN I: INTRODUCCIÓN AL DISEÑO ORIENTADO A OBJETOS. A. Márquez, Lourdes Ortiz, M<sup>a</sup> Pilar Polo, Fco. Roche y Ana M<sup>a</sup> Roldán. Servicio de Publicaciones de la Universidad de Huelva.

COMO PROGRAMAR EN C/C++. H.M. Deitel. Edt. PEARSON Prentice Hall.

C++ ESTÁNDAR. E. Hernández Orallo. Edt. Paraninfo, Thomson Learning.

PROGRAMACIÓN EN C++ PARA INGENIEROS. F. Xhafa, P. Vázquez, J. Marco, X. Molinero y A. Martín. Edt. Thomson.

EL LENGUAJE DE PROGRAMACIÓN C++, B. Stroustrup. Ed. PEARSON Addison Wesley.

PROGRAMACIÓN Y DISEÑO EN C++, J.P. Cohoon, J.W. Davidson. Edt. McGraw-Hill

The C Programming Language by Brian W. Kernighan, Dennis M. Ritchie

Introduction to Algorithms, 3rd Edition, by Cormen, Leiserson, Rivest and Stein.

RESOLUCIÓN DE PROBLEMAS CON C++. W. Savitch. PEARSON Addison Wesley.

## **8. ASSESSMENT**

Divided in two parts:

- Theoretical-Practical Exam (Theory). Consistent in a test where the student will have to solve different problems and/or theoretical questions related to the theoretical topics developed during the quarter. Competences B4 and B5. The Maximum score of this criterion is 50% of the final grade.
- Practical exams in computer science classroom (Practices). Once the laboratory sessions established to work with the practice books of Topics 1, 2, 3 and 4 have been completed, the first practical test will be carried out, in which the knowledge acquired up to that moment will be evaluated. Likewise in the last laboratory session of the quarter, the second test, related in this case with the exercises worked in the practice book of Topic 5. Competences B4 and B5. The maximum score of this criterion is 50% of the final result. The first test will have a weight of 30% of the Practical criterion (1.5 points) and the second of 70% of the Practical criterion (3.5 points).