# ACADEMIC PROGRAM

# INTRODUCTION TO PROGRAMMING I

## B.F.A. IN **COMPUTER SCIENCE**

***MODALITY: ON CAMPUS***

***ACADEMIC YEAR: 2022-2023***

| Name of the course: | Introduction to Programming I |
|---|---|
| Degree : | Computer Science |
| Location: | Centro Universitario de Tecnología y Arte Digital |
| Area: | Fundamentals of Computer Science |
| Year: | 1º |
| Teaching period: | 1 |
| Type: | B |
| ECTS credits: | 6 |
| Teaching modality: | On campus |
| Language: | English |
| Lecturer / Email | Francisco Javier García Algarra / javier.algarra@u-tad.com |
| Web page: | http://www.u-tad.com/ |

## SUBJECT DESCRIPTION

### Area description

This course belongs to the subject “Fundamentals of Computer Science”, where thestudent acquires basic knowledge of the principles that underpin software engineering

### Subject description

The introductory programming course I is the students' first point of contact with the structured creation of programs. For this, C has been chosen as the programming language, due to its efficiency when creating code so that the student can be able to understand the actions that are triggered when writing a line of code. The basic concepts of programming such as data types, operators and expressions and modularity will be reviewed.

## COMPETENCIES AND LEARNING OUTCOMES

### Competencies

BASIC AND GENERAL SKILLS

BC1: Students should demonstrate knowledge in an area of study that builds upon the foundation of general secondary education and goest beyond at a level that, while supported by advanced textbooks, also encompasses certain aspects derived from the cutting edge of their field of study.

BC2: Students should be able to apply their knowledge to their work or vocation in a professional manner, and they should possess the competencies typically demonstrated through the development and defence of arguments as well as problem-solving within their field of study.

BC3: Students must possess the ability to gather and interpret relevant data (usually within their field of study) in order to make judgments that involve reflection on socially, scientifically, or ethically significant issues.

BC4: Students should be capable of conveying information, ideas, problems, and solutions to both specialized and non-specialized audiences.

BC5: Students should have developed the learning skills necessary to pursue further studies with a high degree of autonomy.

CG1 - Ability to understand, plan and solve problems through the development of computing solutions.

GC3 - Knowledge of the scientific fundamentals applicable to the resolution of computing problems.

GC5 - Management of human and technological resources for the correct realization of computer science projects

GC9 - Ability to learn, modify and produce new computer technologies.

SPECIFIC COMPETENCES

CE3 - Knowledge of the relational algebra and the performance of queries in procedural languages for the design of standardized database schemas based on

database schemas based on entity-relational models.

CE9 - Knowledge of control structures, variables, programming syntax and memory usage management in an effective way in the development of a computer application.

in the development of a software application

CE11 - Knowledge of the architecture of the Operating Systems as well as the different mechanisms for the management of processes,

communication and synchronization of processes

CE13 - Knowledge of the fundamentals of computer networks, the different topologies and their communication protocols.

communication protocols

CE16 - Knowledge of the operation of computer systems.

## Learning outcomes

Upon completion of the degree, the graduate will be able to:

- To understand the life cycle of an application using different programming languages

- To know how to debug a software application.

- To know and use markup languages (HTML)
- To build web pages using style sheets (CSS)
- To use the control version tool Git in collaborative development
- To know the fundamentals of Boolean Algebra
- To be able to handle logic gates and simple sequential circuits
- To handle the binary representation of different data types
- To understand von Neumann¿s model
- To understand the hardware architecture of a computer.
- To be able to coda basic programs in assembly language
- To know common network technologies (WiFi, BlueTooth, Ethernet¿)
- To know network topologies
- To understand computer communication using protocols such as ARP, IP, TCP, etc.
- To be able to set basic routing configurations.
- To be able to develop simple network applications
- To know the basic architecture of an Operating System
- To understand the principles of process scheduling
- To understand how the hierarchy of memory works
- To be able to develop a simple file system
- To be able to develop a toy driver
- To understand processes/threads communications and synchronization mechanisms

## CONTENTS

Programming syntax

Standard libraries

Data types, functions and modularity

## SUBJECT SYLLABUS

Topic 1. Introduction

History of computing

von Newman architecture

Hello World!

Topic 2. The C language. Basic elements.

- General structure of a C program
- Elements of a C program
- Compiling a C program
- Variables and Data Types in C
- Constants

Topic 3. Operators and expressions.

- Assignment operator
- Arithmetic operators
- Increment and decrement operators
- Relational operators
- Logical operators
- Bit manipulation operators
- Address operators
- Special operators
- Type Conversions

Topic 4. Selection and control structures

- Control structure
- If with an alternative
- If with two alternatives: if-else
- nested if_else
- Control statement: switch
- while statement
- Repetition: for loop
- Repetition: do .. while loop

Topic 5. Functions

- Concept of a function
- Prototype of a function
- Formal and real parameters
- return instruction

- Passing arguments to functions
- Step by value
- Scope and storage classes
- Recursion

Topic 6. Arrays and chains

- One-dimensional arrays
- Chains
- Multidimensional arrays

Topic 7. Pointers

- Pointer variables
- Pointer operators
- Operations with pointers
- Pointers and arrays
- Features: Pass by reference

## TRAINING ACTIVITIES AND TEACHING METHODOLOGIES

### TRAINING ACTIVITIES

| LEARNING ACTIVITIES | Total hours | Hours of presence |
|---|---|---|
| *Theoretical / Expository classes* | 32,00 | 32,00 |
| *Practical classes* | 22,00 | 22,00 |
| *Tutorials* | 4,00 | 2,00 |
| *Independent study and autonomous work of the student* | 50,00 | 0,00 |
| *Elaboration of work (group or individual)* | 36,00 | 0,00 |
| *Evaluation Activities* | 6,00 | 6,00 |
| *TOTAL* | 150 | 62 |

### Teaching methodologies

Expository method or master lesson

Case learning

Learning based on problem solving

Cooperative or collaborative learning

inquiry learning

Flipped classroom methodology

Gamification

Just in time Teaching (JITT) or classroom on time

Expository method or master lesson

Case method

Learning based on problem solving

Cooperative or collaborative learning

inquiry learning

Flipped classroom methodology

Gamification

## TEMPORAL DEVELOPMENT

DIDACTIC UNITS / TOPICS TIME PERIOD

Topic 1. Programming fundamentals 2 days

Topic 2. The C language. Basic elements 2 days

Topic 3. Operators and expressions 4 days

Topic 4. Selection and control structures 4 days

Topic 5. Functions 4 days

Topic 6. Arrays and chains 6 days

Topic 7. Pointers 6 days

## EVALUATION SYSTEM

| ASSESSMENT SYSTEM | MINIMUM SCORE RESPECT TO THE FINAL ASSESSMENT (%) | MAXIMUM SCORE RESPECT TO THE FINAL ASSESSMENT (%) |
|---|---|---|

| | | |
|---|---|---|
| *Assessment of participation in class, exercises or projects of the course* | 0 | 30 |
| *Assessment of assignments, projects, reports, memos* | 30 | 80 |
| *Objective test* | 10 | 60 |

## GRADING CRITERIA

| ASSESSMENT SYSTEM | ORDINARY EVALUATION | EXTRAORDINARY EVALUATION |
|---|---|---|
| *Assessment of participation in class, exercises or projects of the course* | 5 | 0 |
| *Assessment of assignments, projects, reports, memos* | 40 | 40 |
| *Objective test* | 55 | 60 |

**General comments on the evaluations/assessments**

• The evaluation of participation in class, in practices or in projects of the subject (AE1) will be carried out based on active participation. Active participation will mean carrying out the exercises that are presented daily in class, as well as the correct answers to the questions that are asked. This aspect will represent 5% of the final grade for the subject in the ordinary call. Teachers will require students to hand in some work done in class.

• Throughout the course, mandatory practices or assignments will be proposed that must be delivered before the indicated date through the virtual platform. Once delivered, an evaluable activity will be carried out in class in which certain functions based on the work will be required to be carried out individually. These assignments and activity (AE2) will account for 40% of the final grade for the subject in the ordinary call. The delivery of requested work will be mandatory, and the grade will be obtained from the individual activity carried out in class. Out of form works will not be admitted (guidelines will be given on the form and content of the work delivered), all of them are mandatory to be submitted to be approved in the ordinary call. Likewise, it is mandatory to carry out evaluable activities related to said work. That is, to pass the subject in the ordinary call:

o All mandatory works must be presented, in the requested form/content

• Active participation will be required from the student, necessary for the development of the classes.

• The student will be required to behave well at all times during classes. Bad behavior that prevents the normal development of the class may lead to expulsion from the classroom for a period of time to be determined by the teacher.

o You must have obtained a minimum grade of 5 (out of 10) as an average in the evaluation of the activities of each work.

If the work is not delivered on time, an exceptionally late delivery will be accepted, but the grade for the related activity will have a 30% grade penalty (maximum grade of 7). Summarizing:

- If it is not delivered on time, the activity can be carried out with a 30% penalty

- If it is not delivered, it will be evaluated as a 0. There will be no grade for the activity until this requirement is met. It is mandatory to hand it in to do the media.

- If you do not show up for the activity, it will be evaluated with a 0. The average will continue to be taken with the rest of the activities, but this grade cannot be recovered. In case of justified causes of force majeure, it will be the teacher's power to recover it.

• At the end of the semester, a global exam will be taken, which will have a total weight of 55% on the grade for the ordinary call. To pass the subject in the ordinary session, it will be necessary for the student to have at least a 4 (out of 10) in said exam. Obtaining a 4 on the exam does not mean passing the subject, it only allows you to average the rest of the grades to calculate the final grade. An average will not be taken with less than a 4 on the exam.

• In summary, to pass the subject in the ordinary call:

o It is essential that the final grade (40% of assignments +55% of exam +5% participation) is at least 5.0 (out of 10).

o It is necessary that all mandatory work be delivered

o It is necessary that the average of the activities related to said jobs is at least 5.0 (out of 10)

o It is necessary that the final exam grade be at least 4.0 (out of 10).

• If any of these requirements are not met, the subject will be automatically considered failed regardless of the rest of the grades.

## LIST OF REFERENCES (BOOKS, PUBLICATIONS, WEBSITES):

Basic Bibliography:

David Griffiths, Dawn Griffiths, Head First C, O'Reilly Media, ISBN: 978-1-4493-9991-7.

Luis Joyanes, Fundamentos de programación. McGraw-Hill Interamericana de España S.L. ISBN-13: 978-8448161118.

Bibliografía recomendada:

Brian W. Kernighan, Dennis M. Ritchie, El lenguaje de Programación C. Prentice Hall, ISBN 9789688802052.

Fundamentos de programación utilizando el lenguaje C. Universidad Pontifica de Comillas. ISBN: 978-84-8468-184-7.

"C Programming Absolute beginners' Guide". ISBN: 978-0789751980.

Martín Coin, “Caminando junto al lenguaje C” - Editorial UNRN.

Recommended Bibliography:

Introducción a Computer Science, incluido en la página del profesor Bourke: https://bitbucket.org/chrisbourke/computersciencei/src/master/ComputerScienceOne.pdf.

https://youtu.be/CNFK86hJRfE

https://www.youtube.com/watch?v=ssJY5MDLjlo.

https://ocw.mit.edu/courses/6-s096-introduction-to-c-and-c-january-iap-2013/resources/mit6_s096_iap13_lec1/

## REQUIRED MATERIALS, SOFTWARE AND TOOLS

**Type of classroom**
Theory classroom

Board and projection system

**Materials:**
Personal Computer

**Software:**
Cygwin: Emulador de Linux en Windows

Editor de texto: Notepad ++

Compilador C: gcc