ACADEMIC PROGRAM

INTRODUCTION TO PROGRAMMING II

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

MODALITY: ON CAMPUS

ACADEMIC YEAR: 2022-2023

Name of the course:	Introduction to Programming II
Degree :	Computer Science
Location:	Centro Universitario de Tecnología y Arte Digital
Area:	Programming
Year:	1º
Teaching period:	2
Type:	OB
ECTS credits:	6
Teaching modality:	On campus
Language:	English
Lecturer / Email	Maximiliano Fernández / maximiliano.fernande@u-tad.com
Web page:	http://www.u-tad.com/

SUBJECT DESCRIPTION

Area description

This subject belongs to the programming subject. This subject is dedicated to the study of programming techniques and languages on which the software engineering degree studies will be based.

Subject description

The introductory subject to Programming II completes the training started by students in the first semester and perfects the ability to create structured programs. C has been chosen as the programming language, due to its efficiency when creating code, so the student can be able to understand the actions that are triggered when writing a line of code. Basic programming concepts will be reviewed (not included in Programming I) such as complex data types: structures, unions, enumerations, lists... Concepts for passing parameters, dynamic memory allocation, E file management will be learned. /S and the use of tools for code debugging are introduced.

COMPETENCIES AND LEARNING OUTCOMES

Competencies

BASIC AND GENERAL SKILLSs

CG1 - Ability to understand, schedule and solve problems trough software development

CG3 - Knowledge of the scientific fundamentals applicable to the resolution of computer problems

CG4 - Ability to simplify and optimize computer systems by understanding their complexity

CG9 - Ability to learn, modify and develop new software solutions

CG10 - Use of creative techniques to carry out computer projects

CB1 That students have demonstrated knowledge and understanding in an area of study that starts from the basis of general secondary education, and is usually at a level that, although it is supported by advanced textbooks, also includes some aspects that involve knowledge from the forefront of their field of study.

CB2 Students are able to apply their knowledge to their work or vocation in a professional manner and possess the competences usually demonstrated through the development and defense of arguments and problem solving within their field of study.

CB3 That students have the ability to gather and interpret relevant data (usually within their area of study) in order to make judgements that include reflection on relevant social, scientific or ethical issues.

CB4 Students are able to convey information, ideas, problems and solutions to both specialist and non-specialist audiences.

CB5 That students have developed those learning skills necessary to undertake further studies with a high degree of autonomy.

SPECIFIC SKILLS

CE1 - Knowledge of the structure of computers, the concepts of coding, manipulation, information processing and use of low-level languages

CE7 - Knowledge of the main types of data structures and use of libraries and algorithmic techniques associated with these structures together with the complexity orders that characterize these techniques

CE8 - Conocimiento de los distintos paradigmas detrás de los lenguajes de programación/ Knowledge of the different software

paradigms that underpin programming languages

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

CE10 - Ability to work with a release manager and generate application documentation automatically.

CE15 - Knowledge of fault tolerance, adaptability, load balancing and system predictability for distributed application development

CE17 - Knowledge of the parallelization characteristics of graphics cards and high-performance architectures for application development.

CE20 - Ability to test the operation and functionality of a computer application, develop test plans and use test-oriented design and programming techniques

CE23 - Knowledge of the principles of artificial intelligence and use of deterministic search algorithms and state machines

Learning outcomes

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

- To understand and handle the concept of dynamic memory

- To identify classes with the relevant data of a problem

- To instance classes and objects and manage them

- To understand and use the mechanisms of inheritance, polymorphism and operator overloading

- To identify class relationships in different use-cases.

- To master an object oriented programming language.

- To master programming patterns

- To know different problem solution strategies from an algorithmic view point: divide and conquer, dynamic programming, backtracking or genetic algorithms.

- To understand algorithmic complexity, assess it and search for optimal solutions

- To code a program able to find the optimal path between any pair of nodes of a graph

- To build neural networks to solve applied problems

CONTENTS

Memory management programming

Debugging techniques

SUBJECT SYLLABUS

1. Topic 1 Parameters and Debugging Step

1.1. Pointers Review

1.2. Passing parameters through the command line. argc and argv

1.3. Debug, breakpoints (tools)

2. Topic 2. Structures and unions

2.1. Structures

2.2. Unions

23. Enumerations

2.4. Typedef

3. Topic 3. Dynamic memory allocation

3.1. Introduction

3.2. Memory sections of a program

3.3. malloc() function

3.4. free() function

3.5. calloc() and realloc() functions

4. Topic 4. Chains in depth

4.1. Dynamic string initialization

4.2. Using string functions from the standard library (strcat, strtok,..)

5. Topic 5. Input/Output through text files

5.1. Introduction

5.2. Streams

5.3. Read and write functions

5.4. Binary files in c

5.5. Direct data access

6. Topic 6 Compilation and Complete Projects in C

6.1. Code generation process and creation of an executable

6.2. Complex projects with more than one file

6.3. Library linking and dynamic linking

TRAINING ACTIVITIES AND TEACHING METHODOLOGIES

TRAINING ACTIVITIES

LEARNING ACTIVITIES	Total hours	Hours of presence
Theoretical / Expository classes	35,64	35,64
Practical classes	18,91	18,91
Tutorials	4,00	2,00
Independent study and autonomous work of the student	51,82	0,00
Elaboration of work (group or individual)	33,82	0,00
Evaluation Activities	5,82	5,82
TOTAL	150	62,37

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. Passing Parameters and Debugging Weeks 1-2

Topic 2. Structures and unions Weeks 3-5

Topic 3. Dynamic memory allocation Weeks 6-8

Topic 4. Chains in Depth Weeks 9-10

Topic 5: Input/Output through text files Weeks 11-13

Topic 6 Complete projects in C Weeks 14-15

EVALUATION SYSTEM

ASSESSMENT SYSTEM	MINIMUM SCORE RESPECT TO THE	MAXIMUM SCORE RESPECT

	FINAL ASSESSMENT (%)	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

Ordinary Call

• 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 form/content requested

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.

Extraordinary call

• If the student does not pass the ordinary session, he or she may take the extraordinary session. Namely:

 If the student has failed the subject in the ordinary session due to not having reached 4 in the final exam, but has all his practices delivered and with an average of 5 or higher, then he will have to take the extraordinary exam exclusively and will be kept their notes of the works in ordinary call. His extraordinary exam becomes 60% of the grade and must have a minimum grade of 4 (out of 10) to be averaged with the rest of the grades.

 If the student has failed the subject in ordinary due to pending or failed practices (average less than 5) but his exam grade is greater than or equal to 5, said exam is saved and he will have to present a practice that is indicated at the beginning of the second term. Prior to taking the extraordinary exam, the activity associated with said work will be carried out. The grade for said activity must be at least 5 (out of 10).

 If a 4 has not been achieved in the final exam and there is no average of 5 in the mandatory practices, the student will take a final exam that will represent 60% of their grade for the extraordinary call. The other 40% will be calculated from a deliverable practice that will be presented at the beginning of the second semester. Prior to carrying out the extraordi exam

LIST OF REFERENCES (BOOKS, PUBLICATIONS, WEBSITES):

Basic bibliography

Herbert Schildt

C Reference Manual

Osborne McGraw-Hill

Higher School of Industrial Engineers. university of Navarra

Learn ANSI C language as if you were in First

https://tutoriales.com/aprenda-lenguaje-ansi-c-como-si-estuviera-en-primero/

Mike Banahan, Declan Brady and Mark Doran

The C Book

Addison Wesley

Recommended bibliography

Brian W. Kernighan, Dennis M. Ritchie

The C Programming Language

Pearson-Pretince-Hall

REQUIRED MATERIALS, SOFTWARE AND TOOLS

Type of classroom
Theory classroom

Board and projection system

Materials:
Computer with Windows, macOS or Linux

Software:
Editor de texto (Notepad ++)

Compilador gcc.

Los alumnos disponen del entorno de emulación de Linux sobreWindows cygwin con el compilador gcc integrado y herramienta gdb para depurar