INSTITUTO POLITÉCNICO DE BRAGANÇA

Course Unit	Imperative Programming			Field of study	Computer Science		
Bachelor in	Informatics Engineering			School	School of Technology and Management		
Academic Year	2023/2024	Year of study	1	Level	1-1	ECTS credits	6.0
Type	Semestral	Semester	1	Code	9119-706-1104-00-23		

Workload (hours)	162	Contact hours	T -	TP 60	PL -	TC -	S -	E -	OT -	O -

T - Lectures; TP - Lectures and problem-solving; PL - Problem-solving, project or laboratory; TC - Fieldwork; S - Seminar; E - Placement; OT - Tutorial; O - Other

Name(s) of lecturer(s)	Luís Manuel Alves, Maria João Tinoco Varanda Pereira, Davide Emanuel da Silva Dias, Nelson Alexandre Perdigao Figueiredo, Pedro Gaspar Padrão Antunes Vilares

Learning outcomes and competences

At the end of the course unit the learner is expected to be able to:
1. to develop structured thinking allowing to devise an algorithm, and develop an implementation in C, for computational problems of medium complexity;
2. apply basic knowledge of imperative programming in C, including to structure a program in functions, understand and explore parameters passing and process arrays and strings;
3. apply advanced knowledge of imperative programming in C, namely use pointers and dynamic memory, define and use structures and files, and structure a program in modules.

Prerequisites

Before the course unit the learner is expected to be able to:
Not applicable.

Course contents

Introductory concepts about programming languages; The C language: elementary data types and operations; the If, If-else and Switch selection statements, the While, Do-while and For loops; definition and use of functions; function arguments passed by value; vectors, multi-dimensional arrays and strings; Pointers; using files for input and output; structures, unions and enumerations; definition of new data types; dynamic memory; modularization of programs.

Course contents (extended version)

1. Introductory concepts:
   - computer programming;
   - programming languages;
   - development of a program;
   - the C language.
2. Elementary data types:
   - data types, declaration of variables;
   - concept of constant, definition of symbolic constants;
   - arithmetic operations, statements, assignments, conversions of types;
   - statements to read and write in the console.
3. Testing and conditions:
   - conditions and logical values;
   - logical operators and relational operators;
   - conditional statements If and If-else, Switch statement.
4. Loops:
   - the While statement;
   - the Do-while statement;
   - the For statement.
5. Functions:
   - concept of function and structure of a C function;
   - parameters passed by value;
   - local/global, internal/external and automatic/static variables.
6. Vectors:
   - declaration and automatic initialization of vectors;
   - passing vectors to a function;
   - processing of vectors;
   - multi-dimensional arrays.
7. Strings:
   - main functions for string manipulation;
   - development of specific functions for string manipulation.
8. Pointers:
   - concept of variable, address and pointer;
   - declaration and initialization of pointer variables;
   - operators of pointers;
   - arithmetic of pointers;
   - relationship between pointers and vectors;
   - pointers to pointers.
9. Files:
   - concept of file, peripherals and streams;
   - functions for manipulating files;
   - ways to open a file;
   - reading and writing to text files;
   - reading and writing to binary files;
   - sequential and direct access to files;
   - detection of end of file.
10. Structures, unions and enumerations:
    - concept of structure, declaration and initialization of structures;
    - access to the fields of a structure;
    - passing structures to functions;
    - files of structures;
    - definition of enumerated types;
    - reading and writing of variables of enumerated type;
    - definition of new data types.
11. Dynamic memory allocation:
    - allocation and liberation of memory;
    - functions that return dynamically allocated memory;

Course contents (extended version)

- dynamic data structures;
- implementation and manipulation of linked lists.

12. modularization of programs:
- multi-modular programming concept;
- Concept of prototype of a function;
- implementation of header files.

Recommended reading

1. Luís Damas, “Linguagem C”, Tecnologias de Informação, FCA, 1999.
2. Pedro Guerreiro, “Elementos de Programação com C”, Tecnologias de Informação, FCA, 2006.
3. António Rocha, “Introdução à Programação Usando C”, Tecnologias de Informação, FCA, 2006.
4. Brian W. Kernighan e Dennis M. Ritchie, “The C Programming Language”, Prentice-Hall, 1988.
5. R. Johnsonbaugh, and M. Kalin, “C for Scientists and Engineers”, Prentice-Hall, 1997.

Teaching and learning methods

The teaching method used in lecture classes is the expository method, which makes possible the transmission of knowledge in a continuous and less time consuming manner. Practical classes are mostly based on the active method, enhancing the activity of students through the resolution of practical exercises. Students are also required to perform practical assignments outside the classes.

Assessment methods

1. Alternative 1 - (Regular, Student Worker) (Final)
   - Intermediate Written Test - 30% (First Practical Test solved in the computer.)
   - Intermediate Written Test - 30% (Second Practical Test solved in the computer.)
   - Intermediate Written Test - 40% (Third Practical Test solved in the computer. To be held in the Final Evaluation Period.)
2. Alternative 2 - (Regular, Student Worker) (Supplementary, Special)
   - Final Written Exam - 100%

Language of instruction

1. Portuguese
2. English

Electronic validation

Luís Manuel Alves, Maria João Tinoco Varanda Pereira	Tiago Miguel Ferreira Guimaraes Pedrosa	Luísa Maria Garcia Jorge	José Carlos Rufino Amaro
10-10-2023	25-10-2023	25-10-2023	31-10-2023