

LICENCIATURA: Engenharia Informática	ÁREA CIENTÍFICA: Engenharia Informática
UNIDADE CURRICULAR/CURRICULAR UNIT: Programação I / Programming I	ECTS: 6
DURAÇÃO: Semestral	HORAS DE CONTACTO TEÓRICO PRÁTICAS: 60 (48 TP+12 OT)

OBJETIVOS DE APRENDIZAGEM / LEARNING OUTCOMES OF THE CURRICULAR UNIT

Para concluir com sucesso esta unidade curricular, os estudantes deverão demonstrar possuir os seguintes conhecimentos e capacidades:

1. Conhecer as técnicas fundamentais de programação
2. Transformar problemas propostos em Algoritmos.
3. Transformar Algoritmos em programas escritos em C.
4. Utilizar de forma eficiente a linguagem de Programação C.
5. Utilizar corretamente as ferramentas de edição, compilação e depuração de código
6. Compreender e utilizar funções de manipulação de vetores de caracteres em C (strings);
7. Compreender e utilizar vetores e matrizes;
8. Saber utilizar ponteiros;
9. Formular e Modelar um problema com recurso a estruturas de C (structs)
10. Saber manipular ficheiros;
11. Compreender os conceitos de classes e objetos e respetiva implementação em C++;
12. Saber escrever, compilar executar programas em C++ utilizando classes e objetos.

(English)

To successfully complete this curricular unit, students must demonstrate the following knowledge and skills:

1. Know the fundamental techniques of programming
2. Transform proposed problems into Algorithms.
3. Transform Algorithms into programs written in C.
4. Use efficiently the C programming language.
5. Use correctly the tools for editing, compiling and debugging code
6. Understand and use functions to manipulate strings in C;
7. Understand and use vectors and matrices
8. Know how to use pointers;

9. Formulate and model a problem using C structures (structs)
10. Know how to manipulate files;
11. Understand the concepts of classes and objects and their implementation in C++;
12. Know how to write, compile and execute C++ programs using classes and objects.

CONTEÚDOS PROGRAMÁTICOS / SYLLABUS

1. Algoritmos e Linguagens de Representação Algorítmica
2. A linguagem C
3. Variáveis, Constantes e Tipos Básicos de Dados
4. Elementos da linguagem e comandos
5. Operadores
6. Estruturas de controlo de fluxo da lógica de um programa
7. Funções
8. Estruturas de dados
9. Conceitos de Programação Orientada a Objetos

(English)

1. Algorithms and Algorithmic Representation Languages
2. The C language
3. Variables, Constants and Basic Types of Data
4. Elements of the language and commands
5. Operators
6. Structures of flow control of the logic of a program
7. Functions
8. Data structures
9. Concepts of Object Oriented Programming

DEMONSTRAÇÃO DA COERÊNCIA DOS CONTEÚDOS PROGRAMÁTICOS COM OS OBJETIVOS DA UNIDADE CURRICULAR/ DEMONSTRATION OF THE SYLLABUS COHERENCE WITH THE CURRICULAR UNIT'S OBJECTIVES

Nesta unidade curricular serão ministrados conteúdos que permitam ao estudante conhecer as técnicas fundamentais de programação, nomeadamente com recurso à linguagem C. Para o efeito, far-se-á uma introdução detalhada a técnicas de representação de algoritmos e a respetiva conversão em linguagem C (conteúdos 1 e 2). Serão ministrados conceitos transversais a qualquer linguagem de programação, nomeadamente tipos de variáveis, comandos de entrada e saída de dados, operadores, estruturas de controlo de fluxo e funções (conteúdos 2 a 7). As estruturas de dados (precursor importante para a perceção de classes em POO) serão abordadas em detalhe, bem como um primeiro contacto com técnicas de persistência de informação,

nomeadamente a utilização de ficheiros para um CRUD básico (conteúdos 8 e 9). Será ainda feita uma breve introdução ao paradigma de programação orientada a objetos (conteúdo 10). Para consolidação teórica e prática, será desenvolvida na aula, uma aplicação simples com salvaguarda de informação em ficheiros, com CRUD básico e pesquisa de dados.

(English)

In this curricular unit the students will be taught contents that allow them to know the fundamental techniques of programming, namely using the C language. To this end, it will be made a detailed introduction to techniques of representation of algorithms and their conversion into C language (contents 1 and 2). It will be taught transversal concepts to any programming language, namely variable types, input and output data commands, operators, flow control structures and functions (contents 2 to 7). Data structures (important precursor to the perception of classes in OOP) will be covered in detail, as well as a first contact with information persistence techniques, namely the use of files for a basic CRUD (contents 8 and 9). A brief introduction to the object-oriented programming paradigm will also be made (content 10). For theoretical and practical consolidation, a simple application will be developed in class, safeguarding information in files, with basic CRUD and data search.

METODOLOGIAS DE ENSINO E AVALIAÇÃO / TEACHING METHODOLOGIES INCLUDING EVALUATION

As aulas assumirão um carácter teórico-prático e decorrerão numa sala preparada com software e equipamento informático. Serão lecionadas num contexto baseado em projetos práticos, com demonstrações constantes.

De acordo com o Regulamento de Funcionamento do ISTECS Porto a avaliação é efetuada através de um exame final obrigatório. Na classificação final, poderão ser considerados elementos de avaliação contínua, tais como testes, trabalhos individuais ou em grupo, assim como a participação nas aulas presenciais e com recursos de aprendizagem proporcionados por sistemas de e-learning.

O estudante que realize os trabalhos práticos propostos nas aulas e nas condições aprovadas, poderá prescindir da realização da Prova prática final.

(English)

The classes will assume a theoretical-practical character, and will take place in a room prepared with software and computer equipment. They will be taught in a context based on practical projects, with constant demonstrations.

According to the ISTECS Porto Operating Regulations, the evaluation is carried out through a mandatory final examination. In the final classification, elements of continuous evaluation may be considered, such as tests, individual or group work, as well as participation in face-to-face classes and learning resources provided by e-learning systems.

The student who performs the practical work proposed in the classes and in the approved conditions, may dispense with the completion of the final practical test.

**DEMONSTRAÇÃO DA COERÊNCIA DAS METODOLOGIAS DE ENSINO COM OS OBJETIVOS DA UNIDADE CURRICULAR /
DEMONSTRATION OF THE COHERENCE BETWEEN THE TEACHING METHODOLOGIES AND THE LEARNING OUTCOMES**

A metodologia usada na unidade curricular visa contribuir para que o estudante aprofunde os seus conhecimentos e capacidades no domínio da resolução de problemas, através da elaboração de algoritmos e conversão numa linguagem de programação (no caso, a linguagem C), culminando com o desenvolvimento de trabalhos de grupo e individuais.

Dado ser uma competência prevalente com muita relevância no mercado de trabalho especializado, e pelo facto de introduzir conceitos já relativamente complexos relacionados com técnicas e boas práticas de resolução de problemas e programação, é crítico dotar os estudantes da licenciatura de engenharia informática, do tipo de conhecimentos e competências necessário nestes aspetos. Importa, assim, que as aulas tenham um carácter teórico-prático que garanta o conhecimento sustentado e a sua aplicação concreta em contextos práticos.

(English)

The methodology used in the curricular unit aims to contribute to the students' deepening their knowledge and skills in the field of problem solving, through the elaboration of algorithms and their conversion into a programming language (C language in this unit), culminating with the development of group and individual work.

Given that it is a prevalent competence with much relevance in the specialised job market, and due to the fact that it introduces already relatively complex concepts related to techniques and good practices of problem solving and programming, it is critical to equip students of the computer engineering degree course with the type of knowledge and skills required in these aspects. It is therefore important that the classes have a theoretical and practical character that ensures sustained knowledge and its concrete application in practical contexts.

BIBLIOGRAFIA / BIBLIOGRAPHY

FUNDAMENTAL / ESSENTIAL:

Loureiro, H. (2019). *Guia Moderno de Programação*. Edição FCA

Rocha, A. (2006). *Introdução à Programação Usando C*. Edição FCA

COMPLEMENTAR / COMPLEMENTARY:

Damas, L. (1999). *Linguagem C*. Edição FC

INTERNET:

Acesso a publicações da especialidade, gratuitamente, através da rede SPRINGER:

<https://link.springer.com/>