
1. Caracterização da Unidade Curricular

1.1 Designação

[3655] Programação I / Programming I

1.2 Sigla da área científica em que se insere

IC

1.3 Duração

Unidade Curricular Semestral

1.4 Horas de trabalho

160h 00m

1.5 Horas de contacto

Total: 67h 30m das quais T: 45h 00m | P: 22h 30m

1.6 ECTS

6

1.7 Observações

Unidade Curricular Obrigatória

Unidade Curricular comum ao(s) curso(s) de LEIRT

2. Docente responsável

[697] Jorge Manuel Rodrigues Martins Pião

3. Docentes e respetivas cargas letivas na unidade curricular

[697] Jorge Manuel Rodrigues Martins Pião | Horas Previstas: 67.5 horas

[718] Mário Justiniano Morais Pinheiro | Horas Previstas: 112.5 horas

[1070] Nuno António Afonso Cunha de Oliveira | Horas Previstas: 112.5 horas

4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes)

Os estudantes que terminam com sucesso esta unidade curricular serão capazes de:

1. Demonstrar o conhecimento sobre os mecanismos básicos das linguagens procedimentais.
2. Construir programas de pequena complexidade em linguagem C a partir da sua descrição em linguagem natural.
3. Usar ferramentas para construir, testar e corrigir pequenos programas.



**4. Intended learning outcomes
(knowledge, skills and
competences to be developed
by the students)**

Students who successfully complete this course will be able to:

1. Demonstrate knowledge of the basic mechanisms of procedural languages ξ .
2. Build small complexity programs in C language from its natural language description.
3. Use tools to build, test and debug small programs.

5. Conteúdos programáticos

- Linguagens de programação versus linguagens naturais. Sintaxe e semântica. Algoritmos.
- Tipos de dados elementares. Valores, operadores e expressões. Operações de entrada/saída de dados. Variáveis.
- Instruções de controlo de fluxo: decisão simples e múltipla; ciclos. Funções; Passagem de parâmetros; Alcance e tempo de vida de variáveis.
- Tipos estruturados: *arrays*; *strings*; estruturas; *arrays* multidimensionais
- Algoritmos básicos de pesquisa e ordenação.
- Introdução aos ponteiros. Acesso a ficheiros.
- Introdução à programação modular.

5. Syllabus

- I. Programming languages versus natural languages. Syntax and semantics. Algorithms.
- II. Basic data types. Values, operators and expressions. Input /output operations. Variables.
- III. Flow control instructions: simple and multiple decision; cycles. Functions; Passing parameters; Scope and lifetime of variables.
- IV. Structured types: arrays; strings; structures; multidimensional arrays.
- V. Basic algorithms for searching and sorting.
- VI. Introduction to pointers. File cccessing.
- VII. Introduction to modular programming.

6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular

Esta unidade curricular representa para a maioria dos estudantes o primeiro contacto com a programação, que se pretende motivador sem descurar o formalismo e o rigor, sendo a base da formação de software do curso. São introduzidos conceitos e vocabulário fundamental da programação procedimental (pontos I, II, III e IV do conteúdo programático) concretizados na linguagem C. Os algoritmos de pesquisa e ordenação e o acesso a ficheiros (V, VI) são usados para praticar os conceitos transmitidos. O conceito de ponteiro e a introdução à programação modular (pontos VI e VII) preparam discussões posteriores na área da programação e da arquitetura de computadores.

6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes

This course is for most students a first contact with programming, with motivational purposes, without neglecting formalism and rigor, being the basis of software training course. The fundamental concepts and vocabulary of procedural programming (Sections I, II, III and IV of the syllabus) are presented using the C language. Searching and sorting algorithm text files access, aim to practice the transmitted concepts. Pointers and an introduction to modular programming are introduced in (section VI and VII) prepare subsequent discussions in the area of programming and computer architecture.

**7. Metodologias de ensino
(avaliação incluída)**

Aulas teóricas, com apresentação dos temas, exposição dos conceitos e exemplos de aplicação, complementadas com aulas de prática laboratorial para realização de exercícios de consolidação.

A avaliação tem as componentes escrita (E) e prática (P), ambas pedagogicamente fundamentais.

A componente E é realizada em dois testes (T1 e T2) ou em exame final. T1 é realizado na fase letiva. Na época de exame normal ou de recurso, o estudante pode optar por T2 se obteve o mínimo em T1, ou exame.

A nota mínima em cada teste escrito é de 8 valores, sendo a nota da componente E obtida da média dos testes ou da nota de exame, com o mínimo de 9,5 valores.

A componente P é baseada em três trabalhos práticos, realizados em grupo. Cada trabalho tem a nota mínima de 8 valores.

O conhecimento de cada estudante é avaliado durante a discussão dos trabalhos, sendo atribuídas classificações individuais (mínimo de 9,5 val).

A nota final (C) é obtida através da fórmula: $C = E*0,5 + P*0,5$.

**7. Teaching methodologies
(including assessment)**

Theoretical classes, with presentation of themes, exposition of concepts and application examples, followed by laboratory practice with consolidation exercises. The assessment has written (E) and practical (P) components, both pedagogically fundamental.

Component E can be performed in two tests (T1 and T2) or in a final exam. T1 is carried out in the class phase. In the normal or appeal exam period, student could opt for T2 if he obtained the minimum in T1, or exam.

Minimum score in each written test is 8, with the E component score being obtained by average the tests or the exam score, with a minimum of 9.5.

The P component is based on 3 practical assignments, carried out in groups. Each assignment has a minimum of 8.

Each student knowledge is evaluated during the discussion of the assignments, resulting an individual classification. for the P component (minimum of 9.5).

For the final grade (C) is used the formula: $C = E*0.5 + P*0.5$, in a scale of 0 to 20.

8. Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular

Os conteúdos programáticos são expostos nas aulas teóricas, complementando uma apresentação interativa das matérias com a realização, pelos estudantes, de pequenos exercícios de consolidação das mesmas.

Os objetivos de aprendizagem são desenvolvidos na realização dos trabalhos de grupo.

São efetuadas aulas práticas de acompanhamento dos trabalhos de grupo, que complementam a avaliação, na discussão final, da globalidade dos objetivos de aprendizagem.

8. Evidence of the teaching methodologies coherence with the curricular unit's intended learning outcomes

The syllabus contents are exposed in the theoretical classes, complementing an interactive presentation of the subjects with the realization, by the students, of small consolidation exercises.

Learning objectives are developed in carrying out group work.

Practical classes are carried out to support the group work, which complement the assessment, in the final discussion, of the learning objectives.

9. Bibliografia de consulta/existência obrigatória

K.N.King, *C Programming: a Modern Approach*, 2Ed., W. W. Norton & Company, 2008, ISBN 9780393979503

B. Kernighan, D. Ritchie, *The C Programming Language*, 2^a ed., Prentice Hall, 1988, ISBN 9780131103627 .

10. Data de aprovação em CTC 2024-07-17 2024-07-17

11. Data de aprovação em CP 2024-06-26 2024-06-26