Ficha de Unidade Curricular A3ES
Introdução à Programação
Licenciatura em Engenharia Mecânica
2024-25

1. Caracterização da Unidade Curricular

1.1 Designação
[2070] Introdução à Programação / Introduction to Programming

1.2 Sigla da área científica em que se insere
ECS

1.3 Duração
Unidade Curricular Semestral

1.4 Horas de trabalho
108h 00m

1.5 Horas de contacto
Total: 45h 00m das quais TP: 45h 00m

1.6 ECTS
4

1.7 Observações
Unidade Curricular Obrigatória

2. Docente responsável
[1515] Fernando Paulo Neves da Fonseca Carreira

3. Docentes e respetivas cargas letivas na unidade curricular
[1461] Francisco Mateus Marnoto de Oliveira Campos | Horas Previstas: 315 horas
[1476] Pedro Miguel de Abreu e Silva | Horas Previstas: 90 horas
[1515] Fernando Paulo Neves da Fonseca Carreira | Horas Previstas: 270 horas

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

Conhecer os objetivos da programação e a sua utilização no contexto de engenharia mecânica.

Conhecer diversos tipos de dados e aprender a manipulá-los.

Saber utilizar diversas funções elementares e estruturas de controlo.

Aprender a desenvolver programas de forma estruturada.

Saber desenvolver algoritmos utilizando uma linguagem de programação de algo nível que é amplamente utilizada em diversos contextos.

Tomar conhecimento sobre a implementação de algoritmos numa plataforma de programação no contexto da área de engenharia mecânica.

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

To know the goals of computer programming and its use in the context of mechanical engineering.

To know different data types and to learn how to use them.

To know how to use several elementary functions and control structures.

To develop programs in a structured way.

To know how to develop algorithms using a high-level programming language that is widely used in different contexts.

To learn how to implement an algorithm a computer programming platform in the context of the mechanical engineering.

5. Conteúdos programáticos

1. Introdução à computação. Sistemas de numeração. Representação de dados em computador. Tipos de dados.

2. Ambiente e Sintaxe de linguagem de programação. Operações matemáticas, relacionais, lógicas e alfanuméricas. Estruturas de dados. Variáveis numéricas e alfanuméricas. Listas. Dicionários.

3. Algoritmia. Noções de algoritmo. Estruturas de fluxo: decisão e repetição. Leitura e escrita de ficheiros; Programação modular, utilização de funções e bibliotecas. Implementação em linguagem de programação.

4. Programação por objetos. Noções de objeto, propriedades e métodos. Utilização e criação de objetos.

5. Utilização e desenvolvimento de bibliotecas.

5. Syllabus

1. Introduction to computing. Number systems. Computer data representation. Date types.

2. Programming language environment and syntax. Mathematical, relational, logical and alphanumeric operations. Data structures. Lists. Dictionaries.

3. Algorithmics. The concept of algorithm. Flow structures: decision and repetition. Reading and writing files; modular programming; use of functions and libraries. Implementation in a programming language.

4. Object-oriented programming. Definition of objects, properties and methods. Using and creating objects.

5. Use and development of libraries.

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

O primeiro capítulo dá a conhecer aos alunos a forma como os dados são armazenados nos computadores e como podem ser processados, tendo em conta o tipo de informação.

No segundo capítulo é introduzido o ambiente de programação e são abordados os conceitos base de princípios, ambiente e sintaxe que permitem implementar programas em linguagem de programação.

O terceiro capítulo aborda os conceitos fundamentais de algoritmia, apresentando as principais estruturas de fluxo, as noções de programação modular e a interação com o sistema de ficheiros. Neste capítulo serão implementados algoritmos com vista à resolução de problemas de engenharia.

O quarto capítulo introduz a noção de programação orientada a objetos, muito frequente nas linguagens de programação e necessária para desenvolvimento de alguns algoritmos.

O quinto capítulo dá enfase a outras valências da linguagem de programação, como o desenvolvimento de bibliotecas.

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

The first chapter provides knowledge about how data is stored in computers and how it can be processed considering the data type.

In the second chapter a programming environment is introduced and the basic ideas regarding programming principles, environment and syntax are presented.

In the third chapter the fundamental algorithmics concepts are addressed, along with the introduction of flow structures, modular programming and the interaction with the file system. In this chapter students program several algorithms with the purpose of solving different engineering problems.

The fourth chapter incorporates object-oriented programming, a programming approach accepted in most programming languages and is the mainstay of more complex programs.

The fifth chapter emphasizes other valences of the programming language, such as library development.

7. Metodologias de ensino (avaliação incluída)

A unidade curricular está dividida em aulas teóricas e práticas, sendo 40% das aulas lecionadas em sala de aula e 60% no laboratório.

A avaliação da unidade curricular baseia-se na **avaliação distribuída com exame final.**

Avaliação distribuída: Realização de um trabalho de projeto (**TP**) e de uma prova oral (**PO**), pedagogicamente fundamentais.

Exame Final: Realização de um exame escrito. Na época de exames não há possibilidade de melhoria de nota nem repetição de nenhum componente da avaliação distribuída.

Classificação final: NF = 0,50 Ex + 0,25 TP + 0,25 PO ; mínimo de 9,5 valores para aprovação.

7. Teaching methodologies (including assessment)

The course is composed of theoretical and practical lessons, with 40% of the theoretical classes in the classroom and 60% in the laboratory.

The assessment of the curricular unit is based on **distributed assessment with a final exam.**

Distributed Assessment: Carrying out a project work (**PW**) and an oral exam (**OE**), pedagogically fundamental.

Final Exam: Carrying out a written exam (**Ex**). During exams, there is no possibility for grade improvement or repetition of any component of the distributed assessment.

Final Grade: NF = 0.50 Ex + 0.25 PW + 0.25 OE ; minimum of 9.5 points for approval.

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

Semanalmente, são lecionadas aulas teóricas e práticas no laboratório de informática, permitindo que os alunos desenvolvam programas no computador, em que aplicam os conceitos apresentados na parte teórica.

Após a introdução dos conceitos fundamentais e da sua demonstração na plataforma computacional, com exemplos práticos, são realizados diversos problemas onde os alunos terão de estruturar o seu raciocínio de modo a elaborar algoritmos e a implementá-los, recorrendo à sintaxe da linguagem de programação.

A realização de trabalhos de avaliação permite que os alunos apliquem os conhecimentos teóricos adquiridos, nomeadamente em casos de estudo de engenharia. Os trabalhos envolvem todas as fases de desenvolvimento de uma solução, desde a análise do problema, estruturação dos dados e elaboração do algoritmo, até à implementação numa linguagem de programação.

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

Weekly, the lectures alternate with theorical and practical sessions, conducted in the IT Lab, allowing students to practice the manipulation of variables and algorithmic structures.

After introducing the basic concepts, exercises are performed where students must structure their thinking in order to develop algorithms and implement them using the syntax of the programming language.

The practical assessments evaluate the theoretical knowledge acquired in algorithms and their application in a specific syntax such as: concepts of variable and algorithm, data types, selection and repetition structures, as well as the ability to apply them in developing small applications.

The project allows students to work together developing a computational algorithm with application in the field of Mechanical Engineering. From the problem analysis, data structures and algorithm development to implement in a programming language.

The final exam is an alternative way to continuous evaluation; either for lack of notice required for approval, either by lack of attendance at classes, a factor that impedes the realization of continuous assessment.

9. Bibliografia de consulta/existência obrigatória

Carvalho, Adelaide. 2021. Práticas de Python, Algoritmia e Programação. FCA.

Costa, Ernesto. 2015. Programação Em Python - Fundamentos e Resolução de Problemas. FCA.

Manzano, José Augusto N. G., and Jayr Figueiredo de Oliveira. 2016. Algoritmos - Lógica Para Desenvolvimento de Programação. 28a edição. Érica.

Martins, João Pavão. 2015. Programação Em Python - Introdução à Programação Utilizando Múltiplos Paradigmas. IST.

Neto, João. 2014. Programação Algoritmos e Estruturas de Dados. 3a edição. Escolar Editora.

Norton, P. 2005. Beginning Python. Wiley Pub.

Ramalho, L. 2022. Fluent Python - Clear, Concise, And Effective Programming. O? Reilly Media.

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

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