

Ano Letivo 2019-20

Unidade Curricular INTRODUÇÃO À PROGRAMAÇÃO

Cursos MATEMÁTICA APLICADA À ECONOMIA E À GESTÃO (1.º ciclo)
Tronco comum

Unidade Orgânica Faculdade de Ciências e Tecnologia

Código da Unidade Curricular 18391004

Área Científica INFORMÁTICA

Sigla

Línguas de Aprendizagem Português

Modalidade de ensino Ensino presencial

Docente Responsável Fernando Miguel Pais da Graça Lobo

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
Fernando Miguel Pais da Graça Lobo	PL; T	T1; PL1	30T; 30PL
Docente A Contratar FCT 1	PL	PL2	30PL

* Para turmas lecionadas conjuntamente, apenas é contabilizada a carga horária de uma delas.

ANO	PERÍODO DE FUNCIONAMENTO*	HORAS DE CONTACTO	HORAS TOTAIS DE TRABALHO	ECTS
1º	S1	30T; 30PL	168	6

* A-Anual; S-Semestral; Q-Quadrimestral; T-Trimestral

Precedências

Sem precedências

Conhecimentos Prévios recomendados

Esta cadeira não requer conhecimentos prévios de programação, mas apenas familiaridade com sistemas informáticos.

Objetivos de aprendizagem (conhecimentos, aptidões e competências)

No final desta disciplina, os estudantes aprovados terão demonstrado ser capazes de:

- Compreender a utilização da programação na resolução de problemas do âmbito da sua disciplina de estudos.
- Conhecer a linguagem de programação utilizada.
- Dominar com razoável à-vontade as técnicas de programação elementares.
- Resolver autonomamente problemas de programação simples.
- Apreçar a complexidade algorítmica dos programas que escrevem.
- Identificar as principais componentes dos sistemas computacionais e sua relação com a programação.
- Reconhecer as principais etapas do ciclo de vida do desenvolvimento de software.

Conteúdos programáticos

Introdução à Programação
Elementos constituintes dos programas
Algoritmos básicos
Estruturas de dados fundamentais
Funções
Recursividade
Iteração
Asserções
Buscas e ordenações
Eficiência e ordem de crescimento do tempo de execução
Utilização da memória
Classes
Programação orientada pelos objetos

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

Os conteúdos programáticos correspondem a uma abordagem à programação com ênfase primeiro nas funções e depois nas bibliotecas de funções que equipam a linguagem utilizada. Este ponto de vista, essencialmente intelectual e abstrato, é contrabalançado com necessidade de os programas correrem em sistemas computacionais "físicos", para resolver problemas do "mundo real". Adicionalmente, os estudantes devem ficar equipados com as bases que lhe permitam aprofundar os seus conhecimentos nesta área da informática, por si sós, ou no âmbito de outras disciplinas.

Metodologias de ensino (avaliação incluída)

Nas aulas teóricas, o professor apresenta e discute os temas da disciplina. Nas aulas práticas, os alunos resolvem problemas de programação ou realizam trabalhos mais longos, com guião, no computador. Os alunos completarão a sua formação através de trabalho individual ou em grupo, realizado fora das aulas.

A avaliação usa a modalidade de 'avaliação por frequência', nos termos da alínea b) do número 1 do artigo 9º do Regulamento de Avaliação da UAlg, e consiste num conjunto de trabalhos práticos, cada qual com o seu prazo de entrega e ponderação indicada no enunciado. As soluções submetidas têm de ser posteriormente discutidas individualmente com o professor sob pena de não serem consideradas. O exame assume a forma de uma prova escrita. Só são admitidos a exame os alunos cuja nota de frequência seja maior ou igual a 7,5. A nota final é a média ponderada da nota do exame (70%) com a nota de frequência (30%).

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

As metodologias de ensino correspondem a um modelo tradicional, com aulas teóricas, onde o professor desenvolve a matéria, colocando as ênfases nos pontos mais importantes e realçando as ligações entre os vários tópicos apresentados.

As aulas práticas funcionam num regime de 'laboratório aberto'. Durante as aulas, os estudantes desenvolvem os programas que correspondem aos problemas propostos. Os enunciados desses problemas terão sido publicados previamente, na página Web da disciplina. Assim, idealmente, os estudantes poderão usar a aula para completar os programas que terão começado a resolver em casa. Durante as aulas, os professores supervisionam, aconselham, e esclarecem dúvidas dos alunos.

Bibliografia principal

- John V- Guttag, [Introduction to Computation and Programming Using Python](#) , ISBN-13: 978-0262525008, ISBN-10: 0262525003, MIT Press, 2016.
- Zed A. Shaw, [Learn Python the Hard Way](#) , 3rd edition, ISBN-13: 978-0321884916, ISBN-10: 0321884914, Addison-Wesley, 2014.
- Ernesto Costa, [Programação em Python - Fundamentos e Resolução de Problemas](#) , ISBN: 978-972-722-816-4, FCA, 2015.

Academic Year 2019-20

Course unit INTRODUCTION TO PROGRAMMING

Courses MATHEMATICS APPLIED TO ECONOMICS AND MANAGEMENT
Tronco comum

Faculty / School FACULTY OF SCIENCES AND TECHNOLOGY

Main Scientific Area INFORMÁTICA

Acronym

Language of instruction Portuguese

Teaching/Learning modality Lectures and labs.

Coordinating teacher Fernando Miguel Pais da Graça Lobo

Teaching staff	Type	Classes	Hours (*)
Fernando Miguel Pais da Graça Lobo	PL; T	T1; PL1	30T; 30PL
Docente A Contratar FCT 1	PL	PL2	30PL

* For classes taught jointly, it is only accounted the workload of one.

Contact hours

T	TP	PL	TC	S	E	OT	O	Total
30	0	30	0	0	0	0	0	168

T - Theoretical; TP - Theoretical and practical ; PL - Practical and laboratorial; TC - Field Work; S - Seminar; E - Training; OT - Tutorial; O - Other

Pre-requisites

no pre-requisites

Prior knowledge and skills

This course unit does not require prior programming knowledge but only familiarity with computer systems.

The students intended learning outcomes (knowledge, skills and competences)

At the end of the course, students should be able to:

- Understand the use of programming in the resolution of problems in the scope of the field of study.
- Know the programming language used.
- Master with relative ease the elementary programming techniques.
- Solve individually simple programming problems.
- Assess the algorithmic complexity of the programs produced.
- Identify the main components of computer systems and its relation with programming.
- Recognize the main steps of the life cycle of software development.

Syllabus

Introduction to programming
Basic elements
Notion of algorithm
Fundamental data structures
Functions
Recursion
Iteration
Assertions
Searching and sorting
Efficiency and order of growth of the execution time
Memory usage
Classes
Object oriented programming

Demonstration of the syllabus coherence with the curricular unit's learning objectives

The syllabus corresponds to a programming approach with first emphasis on functions and later on function libraries that equip the language used. This view, essentially intellectual and abstract, is counterbalanced by the need for programs to run on "physical" computer systems to solve "real world" problems. In addition, students should be equipped with the fundamentals that allow them to deepen their knowledge in this field of computer science, either alone or in the context of other disciplines.

Teaching methodologies (including evaluation)

In the lectures, the teacher presents and discusses the topics of the course. In the labs, students solve programming problems. Students complete their training through individual or group work, done outside the classroom.

The evaluation uses the modality of "evaluation by frequency", as prescribed Art.9,1. b), Reg. de Avaliação da UAlg, and consists of a set of programming problems, each with its own deadline and weight which will be specified in the assignment itself. The students' submitted solutions to the programming assignments need to be discussed individually with the teacher, otherwise they won't be considered. The final exam takes the form of written test.

Admission to exam is subject to a frequency grade greater or equal than 7,5. The final grade is the weighted average of the exam grade (70%) and the frequency grade (30%).

Demonstration of the coherence between the teaching methodologies and the learning outcomes

The teaching methodologies correspond to a traditional model, with theoretical classes, where the teacher develops the subject, placing the emphasis in the most important points and highlighting the links between the various topics presented in class.

The practical classes work in an 'open laboratory' regime. During classes, students develop the programs that correspond to the problems proposed. The problem descriptions will have been published previously, in the webpage of the discipline. Thus, ideally, students will be able to use the class to complete the programs they have begun to solve at home. During class, teachers supervise, advise, and help students by answering and their questions and doubts concerning the programming problems.

Main Bibliography

- John V- Guttag, [Introduction to Computation and Programming Using Python](#) , ISBN-13: 978-0262525008, ISBN-10: 0262525003, MIT Press, 2016.
- Zed A. Shaw, [Learn Python the Hard Way](#) , 3rd edition, ISBN-13: 978-0321884916, ISBN-10: 0321884914, Addison-Wesley, 2014.
- Ernesto Costa, [Programação em Python - Fundamentos e Resolução de Problemas](#) , ISBN: 978-972-722-816-4, FCA, 2015.