
Ano Letivo 2019-20

Unidade Curricular LABORATÓRIO DE PROGRAMAÇÃO

Cursos ENGENHARIA INFORMÁTICA (1.º ciclo)

Unidade Orgânica Faculdade de Ciências e Tecnologia

Código da Unidade Curricular 14781068

Área Científica CIÊNCIA DE COMPUTADORES

Sigla

Línguas de Aprendizagem Português-PT

Modalidade de ensino Presencial.

Docente Responsável Pedro João Valente Dias Guerreiro

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
Pedro João Valente Dias Guerreiro	PL; T	T1; PL1	15T; 45PL
João Miguel Gago Pontes de Brito Lima	PL	PL3	45PL
Hamid Reza Shahbazkia	PL	PL4; PL5	90PL
Peter Stallinga	PL	PL2	45PL

* 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	S2	15T; 45PL	168	6

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

Precedências

Sem precedências

Conhecimentos Prévios recomendados

A matéria de Programação Imperativa, cadeira do semente precedente.

Objetivos de aprendizagem (conhecimentos, aptidões e competências)

Desenvolver com autonomia acrescida programas usando a linguagem C.

Dominar com proficiência a linguagem C.

Conhecer os fundamentos da programação gráfica usando Processing.

Compreender o ciclo de desenvolvimento de software.

Descrever o mecanismo de chamada de funções e a pilha de execução.

Explicar a diferença entre a alocação dinâmica de memória e a alocação estática.

Descrever a operação das tabelas de dispersão Saber utilizar estruturas dinâmicas de memória, designadamente as árvores binárias de busca, para implementar conjuntos e tabelas chave-valor.

Exprimir em C e em Processing os principais algoritmos elementares.

Tirar partido das técnicas básicas da programação gráfica em Processing, para desenvolver programas visuais simples, incluindo jogos.

Saber utilizar classes simples, em Processing.

Explicar as principais diferenças entre a programação em C e a programação em Processing.

Conteúdos programáticos

Complementos de programação com C.

Arrays dinâmicos de cadeias de caracteres.

Estruturas e arrays de estruturas.

Arrays a 2 dimensões.

Funções como argumento e funções como valor.

Utilização da função qsort.

Implementação de tabelas de dispersão.

Listas ligadas Árvores binárias de busca.

Introdução à programação gráfica com Processing.

Utilização de classes em Processing.

Programação de jogos.

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

A presente cadeira complementa a cadeira inicial de programação no semestre anterior, introduzindo conceitos fundamentais programação que não couberam na cadeira anterior: alocação dinâmica, arrays a duas dimensões, estruturas de dados recursivas, utilização de funções como argumentos.

Nas segunda parte da cadeira, a programação gráfica em Processing, serve para libertar os alunos dos programas meramente textuais ("entra texto, sai texto"), e, sem sobressaltos, permite introduzir o mecanismo das classes.

Laboratório de Programação contempla exclusivamente tópicos *Core-Tier1* do documento de referência Computer Science Curricula 2013, nas áreas de conhecimento Algoritmos e Complexidade, Linguagens de Programação e, sobretudo, Fundamentos de Desenvolvimento de Software.

Metodologias de ensino (avaliação incluída)

Nas aulas teóricas, o professor discute os temas da cadeira, usando o seu computador para exibir os transparentes, para fazer demonstrações e para ilustrar o desenvolvimento de programas.

Nas aulas práticas, os alunos resolvem pequenos 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 do art. 9, n. 1, al. b) do Regulamento Geral de Avaliação da Universidade do Algarve, de 31 de agosto de 2016. O exame assume a forma de uma prova escrita. Não são admitidos a exame os alunos cuja nota na componente prática seja inferior a um valor fixado no início de cada edição da cadeira (normalmente 7.5).

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

A presente cadeira, Laboratório de Programação, LP, está no segundo semestre do primeiro ano e segue-se a Programação Imperativa. Esta outra cadeira é a cadeira inicial de programação. Nela, os alunos defrontam-se com muitos conceitos e técnicas novos e de natureza diferente do que lhes era habitual. Dificilmente têm tempo para se familiarizam com eles.

Em LP, a componente teórica é menor -- uma hora por semana -- e o ritmo da matéria é mais pausado. Os tópicos são fundamentais, mas requerem já uma certa maturidade.

A componente prática é maior -- três horas por semana -- e dá mais ênfase ao trabalho individual de desenvolvimento de programas de maior dimensão. Isto ajuda os alunos a consolidarem os seus conhecimentos e a ganharem confiança nas suas capacidades.

Na segunda metade da cadeira, os alunos aprendem a programar com Processing. A programação gráfica dá-lhes um alento suplementar no final do ano e permite introduzir as classes, que serão depois usadas abundantemente, no futuro.

Bibliografia principal

The C Programming Language, Dennis M. Ritchie, Brian W. Kernighan, 1988.

Elementos de Programação com C, Pedro Guerreiro, 2006.

Getting Started with Processing, Casey Reas, Ben Fry, 2015.

Apresentações das aulas teóricas, disponíveis na Tutoria Eletrónica.

Academic Year 2019-20

Course unit LABORATÓRIO DE PROGRAMAÇÃO

Courses INFORMATICS (COMPUTER SCIENCE) (1st Cycle)

Faculty / School FACULTY OF SCIENCES AND TECHNOLOGY

Main Scientific Area CIÊNCIA DE COMPUTADORES

Acronym

Language of instruction Portuguese-PT

Teaching/Learning modality Lectures and labs.

Coordinating teacher Pedro João Valente Dias Guerreiro

Teaching staff	Type	Classes	Hours (*)
Pedro João Valente Dias Guerreiro	PL; T	T1; PL1	15T; 45PL
João Miguel Gago Pontes de Brito Lima	PL	PL3	45PL
Hamid Reza Shahbazkia	PL	PL4; PL5	90PL
Peter Stallinga	PL	PL2	45PL

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

Contact hours

T	TP	PL	TC	S	E	OT	O	Total
15	0	45	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

Elementary programming, as taught in the preceding course, Imperative Programming.

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

Develop with increased autonomy programs using the C language.

Master the C language with proficiency.

Know the basics of graphical programming using Processing.

Understand the software development cycle.

Describe the function call mechanism and runtime stack.

Explain the difference between dynamic memory allocation and static allocation.

Describe the operation of the hash tables.

Know how to use dynamic memory structures, namely binary search trees, to implement sets and key-value tables.

Express in C and in Processing the main elementary algorithms.

Use the basic techniques of graphical programming in Processing, to develop simple visual programs, including games.

Know how to use simple classes in Processing.

Explain the main differences between C programming and Programming in Processing.

Syllabus

Complements of programming with C.

Dynamic string arrays.

Structures and arrays of structures.

Arrays in 2 dimensions.

Functions as arguments and functions as values.

Using the qsort function.

Implementation of hash tables.

Linked Lists.

Binary search trees.

Introduction to graphical programming with Processing.

Use of classes in Processing.

Game programming.

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

The present course complements the initial programming course in the previous semester, introducing fundamental programming concepts that did not for which there was no time or opportunity in that previous course: dynamic allocation, 2-dimensional arrays, recursive data structures, use of functions as arguments.

In the second part of the course, graphical programming in Processing frees the students from the merely textual programs ("text in, text out"), and allows us to swiftly introduce the mechanism of the classes.

The course Programming Laboratory covers only "Core-Tier1" topics in the Computer Science Curricula 2013 reference document, in the areas of Algorithms and Complexity, Programming Languages, and Software Development Fundamentals.

Teaching methodologies (including evaluation)

In the lectures, the teacher discusses the topics of the course, using his computer to display PowerPoint slides, to demonstrate programs being tested and to illustrate the development of programs.

In the labs, students solve small programming problems or perform longer tasks following detailed scripts provided.

Students complete their study through individual or group work, done outside of class.

The evaluation uses the "evaluation by frequency modality", according to the terms of art. 9, n. 1, al. b) of the General Evaluation Regulation of the University of Algarve, of August 31, 2016. The examination takes the form of a written test. Students whose scores in the practical component are less than a value fixed at the beginning of each edition are not admitted to the examination.

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

The present course, Programming Laboratory, LP, is in the second semester of the first year and follows the Imperative Programming. This other course is the initial course in programming. In that course, students are confronted with many new concepts and techniques, of a different nature from what they used to. They barely have time to get acquainted with them, and to practice a little.

In LP, the theoretical component is smaller - one hour per week - and the rhythm of the matter is more leisurely. The topics are fundamental, but they do require some maturity.

The practical component is larger - three hours per week - and puts more emphasis on individual work on developing larger programs. This helps students to consolidate their knowledge and gain confidence in their abilities.

In the second half of the course, students learn to program with Processing. The graphical programming gives them an extra boost at the end of the school year and allows to introduce the classes, which will then be used abundantly in the future.

Main Bibliography

The C Programming Language, Dennis M. Richie, Brian W. Kernighan, 1988.

Elementos de Programação com C, Pedro Guerreiro, 2006.

Getting Started with Processing, Casey Reas, Ben Fry, 2015.

Lectures slides, available at the learning management system.