

F10. FICHA DE UNIDADE CURRICULAR - R5/R6

Instituição:	Instituto Superior de Engenharia do Porto			
Curso:	Licenciatura em Engenharia Informática			
Disciplina/Unidade:	Paradigmas da Programação			

I - IDENTIFICAÇÃO				
Ano Escolar - 2013-2014		Área Temática/Grupo * - C		Código Interno - PPROG
ANO:	TRI/SEMESTRE:	TEMPO DE CONTACTO - HORAS/SEMANA:	ECTS:	NÍVEL (B/I/A)**:
1	2º Semestre	T: 1; TP: 1.5; PL: 4	6	
Pré-requisitos formais (Precedências):				
Não tem				
Endereço da Página da Disciplina na Internet: https://moodle.isep.ipp.pt				
Outro:				
Docente				
	Nome	Posição	Formação Académica	% Ocupação
Responsável	Luiz Felipe Rocha De Faria (LEF)	Professor Adjunto	Doutoramento	
	Alvaro Jose Carvalho Teixeira (ACT)	Professor Adjunto	Mestrado	
	António Alexandre De Sousa Gouveia (AAS)	Equiparado Assistente 2º Triénio D/M	Mestrado	
	Emanuel Fernando Da Cunha Silva (ECS)	Equiparado Assistente 2º Triénio D/M	Mestrado	
	Fernando Jorge Ferreira Duarte (FJD)	Professor Adjunto	Doutoramento	
Outros	José Marilí Oliveira Cardoso (JOC)	Equiparado Assistente 2º Triénio	Licenciatura	
	Luís Andre Andrade Da Silva Oliveira (LAO)	Equiparado Assistente 1º Triénio	Licenciatura	
	Maria Dulce Fernandes Mota (MDM)	Professor Adjunto	Mestrado	
	Nelson Manuel Faria Freire (NMF)	Assistente 2º Triénio	Licenciatura	
	Nuno Miguel Vieira Morgado (NVM)	Equiparado Assistente 2º Triénio	Licenciatura	
	Paula Correia Tavares (PCT)	Equiparado Assistente 2º Triénio	Licenciatura	
	Piedade Barros Lopez Carvalho (PBC)	Professor Adjunto	Doutoramento	

* Ciências Básicas (B), Ciências de Engenharia(C), Disciplinas da especialidade (S), Disciplinas opcionais (O), Temas complementares (P).
 ** Básico/Intermédio/Avançado.
 *** Percentagem da unidade curricular no total da atividade lectiva do docente no semestre.

II - PROPÓSITOS, RESUMO, CARATERIZAÇÃO	
Antecedentes e enquadramento (max. 600 caracteres)	
Apresentar uma perspetiva dos aspectos técnicos e científicos que justificam esta unidade, bem como o seu enquadramento no curso.	
Português	Inglês
O paradigma de programação orientado ao objecto surgiu para aumentar a reutilização e facilitar a manutenção de código, sendo actualmente o paradigma de programação mais utilizado. Esta unidade curricular está inserida no segundo semestre do primeiro ano e tem por objectivo transmitir os conceitos da programação orientada ao objecto. Espera-se que os alunos tenham adquirido previamente conhecimentos sólidos acerca dos tópicos ensinados na unidade curricular de Algoritmia e Programação (APROG).	The object-oriented programming paradigm was developed to increase the reusability and maintainability of source code, being presently the most used programming paradigm. The PPROG course is integrated in the second term of the first year and aims to convey the underlying concepts of object-oriented programming paradigm. It is expected that students have acquired a good knowledge of topics taught in the course of Algoritmia e Programação (APROG), a course from the first term of the first year.
Propósitos e objetivos gerais (max. 750 caracteres)	
Nomear os objetivos pedagógicos e qual a contribuição desta unidade para o curso. Podem ser apresentados objetivos gerais.	
Português	Inglês
O aluno deve: - Desenvolver um entendimento acerca dos princípios subjacentes à programação orientada ao objecto - Aplicar abordagens baseadas na metáfora do objecto usando a linguagem de programação Java.	The student should: - To develop an understanding of the principles underpinning object oriented programming - To apply object-based approaches using the Java programming language.
Propósitos e objetivos específicos (max. 1000 caracteres)	
Nomear os objetivos pedagógicos específicos, sendo que para estes últimos deve ser apresentada a correspondência com Syllabus do curso (caso exista).	
Português	Inglês
O aluno deve ficar apto a: 1. Explicar a motivação para o desenvolvimento baseado em linguagens de programação orientadas ao objecto (1.2.2, 2.1.2) 2. Descrever as características principais de uma linguagem de programação orientada ao objecto (1.2.1) 3. Produzir e/ou fazer a depuração de fragmentos de código que apliquem os princípios do desenvolvimento de software orientado ao objecto (1.2.1, 1.2.2) 4. Desenvolver as competências para a implementação de aplicações orientadas ao objecto que apresentem características de modularidade, extensibilidade, robustez e generalidade (1.2.2, 2.1.2, 2.1.3, 4.4.1) 5. Entender e aplicar os princípios da programação gerida por eventos em programas dotados de interface gráfica com o utilizador. (1.2.2)	The student should be able to: 1. Explain the motivation for the development based on object oriented programming languages (1.2.2, 2.1.2) 2. Describe the essential features of an object oriented programming language (1.2.1) 3. Produce and/or debug code fragments that illustrate principles of object oriented software development (1.2.1, 1.2.2) 4. Develop the ability to develop object-oriented applications with features of modularity, extensibility, robustness, and generality (1.2.2, 2.1.2, 2.1.3, 4.4.1) 5. Understand and apply event-driven programming principles in graphical user interface (GUI) programs. (1.2.1)
Conteúdo/Programa (max. 1000 caracteres)	
Descrever o programa teórico e prático adequado indicando o peso relativo dos itens do programa.	
Português	Inglês
1. Objects and Classes(1.5weeks) Object-Oriented Programming Instance variables Public and private modifiers Public interface of a class Instance methods Constructors Overloading Object references Static variables and static methods 2 Inheritance, Polymorphism and Interfaces(2weeks) Inheritance Subclasses and superclasses Substitution principle	1. Objects and Classes(1.5weeks) Object-Oriented Programming Instance variables Public and private modifiers Public interface of a class Instance methods Constructors Overloading Object references Static variables and static methods 2 Inheritance, Polymorphism and Interfaces(2weeks) Inheritance Subclasses and superclasses Substitution principle

Substitution principle		Substitution principle	
Calling methods and constructors of a superclass		Calling methods and constructors of a superclass	
Overriding		Overriding	
Polymorphism		Polymorphism	
Abstract classes		Abstract classes	
Final methods and classes		Final methods and classes	
Protected modifier		Protected modifier	
Object class		Object class	
Interface types		Interface types	
Comparable native interface		Comparable native interface	
3 Object-Oriented Design(1week)		3 Object-Oriented Design(1week)	
Classes and their responsibilities		Classes and their responsibilities	
Relationships between classes		Relationships between classes	
Packages		Packages	
4 Array Lists(0.5weeks)		4 Array Lists(0.5weeks)	
5 Exception handling(0.5weeks)		5 Exception handling(0.5weeks)	
6 Graphical user interfaces and event handling(1.5weeks)		6 Graphical user interfaces and event handling(1.5weeks)	
Frame windows		Frame windows	
Events and event handling		Events and event handling	
Processing text input		Processing text input	
Introduction to layout management		Introduction to layout management	
Mouse events		Mouse events	
7 Streams, binary input/output and serialization(1week)		7 Streams, binary input/output and serialization(1week)	
8 Java collections(1.5weeks)		8 Java collections(1.5weeks)	
9 Generic classes and Enum types(1.5weeks)		9 Generic classes and Enum types(1.5weeks)	
Material de ensino mais importante			
Nomear os livros de texto principais e outros textos/materiais de suporte básicos.			
Português		Inglês	
- Apontamentos de acompanhamento das aulas (disponíveis no Moodle no endereço http://moodle.isep.ipp.pt)		- Lecture notes (available in Moodle at http://moodle.isep.ipp.pt)	
- Horstmann, Cay. Big JAVA Late Objects (5th Edition), John Wiley & Sons, Inc., ISBN 978-1-118-08788-6		- Horstmann, Cay. Big JAVA Late Objects (5th Edition), John Wiley & Sons, Inc., ISBN 978-1-118-08788-6	
Material de ensino complementar			
Livros de referência complementares, artigos e materiais com informação técnica apropriada.			
Português		Inglês	
- Bloch, Joshua. Effective Java (2nd Edition), Addison-Wesley, ISBN 978-0-321-35668-0		- Bloch, Joshua. Effective Java (2nd Edition), Addison-Wesley, ISBN 978-0-321-35668-0	
- Java Platform Documentation: http://docs.oracle.com/javase/7/docs/		- Java Platform Documentation: http://docs.oracle.com/javase/7/docs/	
Ferramentas de ensino/aprendizagem (se aplicável)			
Software, ferramentas de e-learning, etc.			
Português		Inglês	
- Moodle ISEP (https://moodle.isep.ipp.pt)		- Moodle ISEP (https://moodle.isep.ipp.pt)	
- Netbeans IDE 7.4		- Netbeans IDE 7.4	
- Git version control system and Bitbucket repository (www.bitbucket.org)		- Git version control system and Bitbucket repository (www.bitbucket.org)	
Conhecimentos prévios assumidamente adquiridos			
Enumerar (descrever em linhas os conhecimentos esperados) e identificar a fonte (identificar as unidades onde ele foram transmitidos).			
Português		Inglês	
Pressupõem-se que os alunos estejam familiarizados com a linguagem de programação Java. Os alunos devem possuir um bom nível de conhecimentos acerca das matérias abordadas na unidade curricular de Algoritmia e Programação (APROG), que funciona no primeiro semestre do primeiro ano.		Students are expected to be familiar with the Java programming language. Students must have good knowledge about topics taught in the course of Algorithms and Programming (APROG), a course from the first term of the first year.	
Metodologia Ensino/Aprendizagem descrevendo, nomeadamente, os métodos inovadores (max. 500 carateres)			
Apresentar as estratégias metodológicas (métodos e técnicas pedagógicas) e atividades de treino, realçando os aspectos inovadores. Caso haja diferenciação de metodologia de aprendizagem entre turmas, esta deve ser explicada.			
Português		Inglês	
Nas aulas T, apresentações orais são complementadas por discussões para estimular a aprendizagem activa. Nas aulas TP são usados casos de estudo para ilustrar os conceitos e técnicas apresentadas nas aulas T. Nas aulas PL os alunos são convidados a executar tarefas de programação que requerem a compreensão dos tópicos apresentados nas aulas T e TP. Os alunos desenvolvem um projecto elaborado por grupos de dois estudantes, contando com a supervisão dos professores.		In T lectures, oral presentations are complemented with discussions to stimulate active learning. In TP classes, case studies are presented and discussed with students. The aim of these case studies is to provide contexts to illustrate the application of concepts and techniques presented in T lectures. In PL classes, students are invited to develop programming tasks that require them to understand the topics presented in lectures. The project is developed by groups of 2 students with supervision	
Caraterização dos objectivos e programa			
A - Distribuição percentual estimada do conteúdo científico e tecnológico			
Componente científica (estabelece e desenvolve bases científicas)		80	
Componente tecnológica (aplicada ao projeto e funcionamento dos processos)		20	
Contexto envolvente		0	
Caraterização dos objetivos e programa (max. 1000 carateres)			
B - Resultados expetáveis (outcomes) - em conformidade com os critérios EUR-ACE			
Descrever o que é esperado que o estudante compreenda, saiba ou seja capaz de realizar depois de frequentar esta unidade curricular relativamente aos seis "outcomes" das orientações do EUR-ACE: conhecimento e compreensão, análise em Engenharia, projecto em Engenharia, prática em Engenharia, investigação e contexto envolvente.			
Português		Inglês	
#Conhecimento e compreensão		#Knowledge and Understanding	
- Conhecimento acerca dos conceitos principais da programação orientada ao objecto. Compreensão acerca da organização do código em classes e respectivas vantagens. (1.2.1)		- Knowledge about the main concepts of object-oriented programming. Understanding the organization's code of a program into classes and their benefits. (1.2.1)	
#Análise em Engenharia		#Engineering Analysis	
- Análise dos aspectos relacionados com o projecto de classes considerando as propriedades de modularidade, encapsulamento, composição, herança e polimorfismo. (1.2.1, 1.2.2)		- Analysis of issues related to the design of classes aimed at modularity, encapsulation, composition, inheritance, and polymorphism. (1.2.1, 1.2.2)	
#Projecto em engenharia		#Engineering Design	
- Competência para desenvolver aplicações orientadas ao objecto apresentando as características de modularidade, extensibilidade, robustez e generalidade. (1.2.2, 2.1.2, 2.1.3)		- Ability to develop applications with object oriented features of modularity, extensibility, robustness and generality. (1.2.2, 2.1.2, 2.1.3)	
#Investigação		#Research	
N.A.		NA	
#Prática em engenharia		#Engineering Practice	
- Capacidade para desenvolver software modular, reutilizável e escalável. (4.4.1)		- Capacity to develop modular, reusable, and scalable software. (4.4.1)	
#Contexto envolvente		#Context/Transferable Skills	
- Compreensão da transição a partir de programação de pequena escala para problemas de programação em larga escala, adoptando as técnicas de		- Understanding the transition from small-scale programming to programming problems in large-scale, adopting the techniques of object-oriented	

problemas de programação em larga escala, adoptando as técnicas de programação orientada ao objecto. (4.4.1)	programming. (4.4.1)
--	----------------------

III - PROCEDIMENTOS DE AVALIAÇÃO

Auto-avaliação (se aplicável)

Ferramentas que permitam aos alunos a avaliação autónoma das competências adquiridas.

Português	Inglês
N.A.	NA

Avaliação pela docência

Modelo de avaliação das competências da unidade curricular.

Tipo(s) de avaliação permitido(s)

☒ Avaliação durante o período letivo com avaliação final

Tipo de avaliação predefinido

(aplicável a estudantes que não escolham tipo de avaliação)

Avaliação durante o período letivo com avaliação final

Avaliação durante o período letivo com avaliação final (max. 200 caracteres)

Os estudantes têm a possibilidade de realizar parte da avaliação durante o período de aulas.

Neste caso, os Estudantes terão acesso a todas as épocas de exame, se necessário e verificadas as respetivas condições de acesso definidas na FUC. A avaliação durante o período letivo não deverá ter um peso inferior a 50% da classificação final. Exceções a este valor, até um mínimo de 30%, deverão ser devidamente fundamentadas, mediante parecer positivo do Diretor de Curso, ao Conselho Pedagógico.

Número de momentos de avaliação durante o período letivo: 2

Português	Inglês
NFREQ=0.4*TPI + 0.6*TPG, se NFREQ < 8,0 => SMNF Nota final=0.5*NFREQ+0.5*ExameFinal, se ExameFinal < 8,0 => SMR TPI: trabalhos individuais TPG: projecto (grupos de 2 alunos)	NFREQ=0.4*TPI + 0.6*TPG, if NFREQ < 8,0 => SMNF Final grade=0.5*NFREQ+0.5*FinalExam, if FinalExam < 8,0 => SMR TPI: individual homework TPG: project (groups of 2 students)

Avaliação em recurso

Português	Inglês
Na época de recurso apenas é possível repetir o exame final, mantendo todas as outras componentes os seus pesos relativos. NFREQ=0.4*TPI + 0.6*TPG, se NFREQ < 8,0 => SMNF Nota final=0.5*NFREQ+0.5*ExameFinal, se ExameFinal < 8,0 => SMS	It is only possible to repeat the exam, keeping all other components their relative weights. NFREQ=0.4*TPI + 0.6*TPG, if NFREQ < 8,0 => SMNF Final grade=0.5*NFREQ+0.5*FinalExam, if FinalExam < 8,0 => SMS

Melhoria de Nota de Avaliação

Português	Inglês
Existem duas possibilidades para melhoria de classificação: a) Repetir apenas o exame final, mantendo as componentes obtidas anteriormente. b) Melhorar todas as componentes de avaliação através da inscrição e frequência em nova edição da unidade curricular.	There are two possibilities: a) Retake the final exam only, keeping the other components obtained previously. b) Try to improve all components by re-enrolment in a new edition.

Frequência anterior aceite Não

Nota biográfica do responsável da Unidade Curricular

Português	Inglês
O docente responsável pela disciplina possui o doutoramento em Engenharia Electrotécnica e de Computadores e exerce as funções de Professor Adjunto do Departamento de Engenharia Informática. A sua actividade de Investigação e Desenvolvimento incide na área da Inteligência Artificial, em particular na área dos Sistemas Baseados em Conhecimento. Actualmente exerce as funções de subdirector do Departamento de Engenharia Informática do ISEP.	The teacher has a PhD in Electrical Engineering and is Auxiliary Professor of the Department of Informatics Engineering. He develops his research activities in the Artificial Intelligence domain, in particular in the field of Knowledge Based-Systems. Actually he is vice-director of the Department of Informatics Engineering in ISEP.

Observações

Português	Inglês
<p>#Avaliação durante o período lectivo com avaliação final</p> <p>Durante o semestre (NFREQ):</p> <p>a) Trabalhos individuais (TPI, 40%)</p> <p>No final do semestre (semana 12):</p> <p>b) Projecto (grupos de 2 alunos) (TPG, 60%)</p> <p>NFREQ = 0.4*TPI + 0.6*TPG, Se NFREQ < 8.0 => SMNF</p> <p>Exame final:</p> <p>c) Exame, se EX < 8.0 => SMR</p> <p>Classificação final = 0.5*NFREQ + 0.5*EX</p> <p>As datas limite para submissão de trabalhos devem ser respeitadas. Os elementos para avaliação entregues após as datas limite poderão ser rejeitados ou sujeitos a penalização na avaliação, de acordo com as instruções disponibilizadas no Moodle</p>	<p>#Assessment during the semester and final exam</p> <p>During the semester (NFREQ):</p> <p>a) Individual homework (TPI, 40%)</p> <p>At the end of the semester (week 12):</p> <p>b) Project (groups of 2 students) (TPG, 60%)</p> <p>NFREQ = 0.4*TPI + 0.6*TPG, if NFREQ < 8.0 => SMNF</p> <p>Final exam:</p> <p>c) Exam, if EX < 8,0 => SMR</p> <p>Final grade = 0.5*NFREQ + 0.5*EX</p> <p>All deadlines are to be honored. Late delivery of assignments/components may be impossible or penalized, according to the instructions available on moodle.</p>

Instruções disponibilizadas no Moodle.

#Código de conduta:

- Nenhum aluno deve assumir a posse de trabalho que não seja da sua autoria.
- É expressamente proibido o uso de código e/ou diagramas não referenciados de outros alunos ou de outras fontes (por exemplo, Internet, etc.).
- Código proveniente de fontes externas deve estar claramente identificado, incluindo a identificação do seu autor e respectiva fonte.
- Casos de uso abusivo de código (sem o consentimento do autor) serão reportados ao Conselho Pedagógico.

#Code of conduct:

- No group should assume ownership of work that was not his own.
- Is strictly forbidden the use of unreferenced code and/or diagrams of other groups or other sources (e.g. Internet, etc.).
- Code from external sources must be clearly identified in the code itself, indicating the author and / or source.
- In case of detection of code sharing between groups, all groups involved will be penalized.
- Cases of theft of code (without the cooperation of the author) will be reported to the Pedagogic Council.

Data/hora de aprovação da ficha de unidade curricular: 2014-02-26 18:35.