

Ano Lectivo	2016/17																									
Curso	Engenharia Informática																									
Unidade Curricular	Programação Orientada a Objectos																									
Língua de ensino	Português																									
ECTS/tempo de trabalho (horas)	<table border="1"> <thead> <tr> <th rowspan="2">ECTS</th> <th rowspan="2">Total</th> <th colspan="7">Horas de contacto semestral</th> </tr> <tr> <th>T</th> <th>TP</th> <th>PL</th> <th>S</th> <th>TC</th> <th>O</th> <th>OT</th> </tr> </thead> <tbody> <tr> <td>5</td> <td>135</td> <td>30</td> <td>15</td> <td>15</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p>T - Teóricas; TP - Teórico-práticas; PL - Prática-laboratorial; S - Seminário; OT - Orientação tutorial; TC - Trabalho de campo; E - Estágio; O* - Outras horas caracterizadas como Ensino Clínico ao abrigo da Diretiva nº 77/453/CEE de 27 Junho adaptada pela Diretiva 2005/36/CE;</p>	ECTS	Total	Horas de contacto semestral							T	TP	PL	S	TC	O	OT	5	135	30	15	15				
ECTS	Total			Horas de contacto semestral																						
		T	TP	PL	S	TC	O	OT																		
5	135	30	15	15																						
Docente Responsável/Carga letiva <small>[nome completo e e-mail]</small>	Luís Manuel Tremoceiro Baptista / lmtb@estgp.pt																									
Outros Docentes e respetivas cargas letivas <small>[nome completo e e-mail]</small>	Sérgio Duarte Correia / scorreia@estgp.pt																									
Pré-requisitos <small>[competências à entrada; pré-requisitos; precedências]</small>	Não																									
Objetivos da aprendizagem (conhecimentos, aptidões e competências) a desenvolver pelos estudantes, operacionalização dos objetivos e medição do seu grau de cumprimento)	Compreender os princípios da linguagem orientada a objetos. Criar competência na programação em linguagem orientada a objetos (C++). Criar competências de análise orientada a objetos (UML).																									
Conteúdos Programáticos <small>[estrutura de conteúdos a desenvolver para o total de horas previsto]</small>	Introdução à linguagem orientada a objetos. Conceitos da linguagem orientada a objetos: Abstração; Encapsulamento; Hierarquia; Objetos; Classes. Relações entre Classes: Diagrama de classes; Associação; Agregação; Composição; Herança. Programação em C++: Construção de Classes; Overload de funções e operadores; Reutilização de classes; Herança e Polimorfismo, Classes abstratas; Alocação dinâmica de objetos; Templates; Tratamento de exceções.																									
Demonstração da coerência dos conteúdos programáticos com os objetivos da unidade curricular	Os conteúdos que incluem os conceitos da linguagem orientada a objetos e da relação entre classes, permitem ao aluno perceber o paradigma das linguagens orientadas a objetos, e por isso atingir o objetivo identificado como compreender os princípios da linguagem orientada a objetos. Por outro lado, a relação entre classes é um conteúdo transversal à unidade curricular, utilizado sempre que é necessário apresentar ou construir modelos de relações entre classes. Isto permite criar competências de análise orientada a objetos. Finalmente, as competências relacionadas com a programação orientada a objetos são atingidas com os conteúdos da programação em C++, que abordam as técnicas usadas em programação orientada a objetos.																									
Metodologias de ensino (avaliação incluída) <small>[indicar os produtos, critérios e pesos de avaliação] (máx1000 caracteres)</small>	<p>1 - Metodologias de ensino</p> <p>Aulas teóricas com exposição da matéria e apresentação de exemplos/casos de estudo. Aulas teórico-práticas com elaboração de exercícios de programação.</p> <p>2 - Avaliação por frequência</p> <p>Teste escrito (40%), com toda a matéria teórica. O teste tem nota mínima de 8. Trabalhos práticos de programação (20% + 40%). Os trabalhos têm nota mínima de 10.</p>																									

	<p>Estes trabalhos são feitos individualmente ou em grupo de 2 elementos. Um primeiro trabalho (20%) inclui apenas a parte inicial da matéria e tem como objetivo treinar o aluno na programação OO. O segundo (40%) inclui toda a matéria.</p> <p>Os trabalhadores estudantes são dispensados das aulas práticas, mas ficam obrigados a realizar os trabalhos práticos.</p> <p>3 - Avaliação por Exame</p> <p>Igual à avaliação de frequência.</p> <p>O aluno pode aproveitar os trabalhos práticos das avaliações anteriores, desde que atinjam a nota mínima.</p>
<p>Demonstração da coerência das metodologias de ensino com os objectivos da aprendizagem da unidade curricular</p>	<p>As aulas teóricas pretendem introduzir os conceitos da programação orientada a objetos e apresentar exemplos/casos de estudo que ilustrem esses conceitos. Esta parte teórica é complementada com uma parte prática onde os alunos vão implementar exercícios relacionados com aqueles conceitos. No final do semestre os alunos terão compreendido os princípios das linguagens orientadas a objetos. A parte prática permite iniciar o aluno na análise e na programação orientada a objetos.</p> <p>Os trabalhos práticos permitem implementar problemas mais complexos e exigentes, que não seriam possíveis implementar nas aulas. Estes trabalhos práticos aprofundam e cimentam as competências relacionada com a programação e com a análise orientada a objetos, nomeadamente o segundo, que é um trabalho de maior dimensão, e que inclui toda a matéria.</p> <p>Uma vez que o cumprimento dos objetivos desta unidade exige uma forte componente prática de programação optámos por dar mais importância à parte prática na avaliação final. Assim, a parte dos trabalhos práticos representa 60% da nota final, uma vez que é nesta parte que o aluno demonstra que atingiu os objetivos da unidade. No entanto, a avaliação por frequência (40%) permite avaliar, de forma mais precisa, o nível de conhecimento do aluno.</p>
<p>Bibliografia Principal</p>	<p>Lippman, S., Lajoie, J e Moo, B., "C++ Primer", Addison Wesley Professional, 4ª Edição, 2005.</p> <p>Eckel, B., "Thinking in C++ - Volume 1", 2ª Edição, http://www.bruceeckel.com/, 2000.</p> <p>Eckel, B. e Allison, C., "Thinking in C++ - Volume 2", http://www.bruceeckel.com/, 2003.</p> <p>Stroustrup, B., "The C++ Programming Language", Addison Wesley Professional, Special Edition, 3ª Edição, 2000.</p> <p>Nicolai M. Josuttis, "C++ Standard Library, The: A Tutorial and Reference", Addison Wesley Professional, 1999.</p> <p>Nunes, M., O'Neil, H., "Fundamental de UML", Coleção Fundamental, FCA — Editora de Informática, Lisboa, 2001.</p> <p>Booch, G., "Object-Oriented Analyses and Design with Applications", The Benjamin/Cummings Publishing Co, Inc, 2º Edition, 1994.</p> <p>Kernighan, B. e Ritchie, D., "The C Programming Language", Second Edition Prentice Hall, 1988.</p>
<p>Bibliografia Complementar</p>	
<p>Situações especiais [estudantes com estatuto especial]</p>	<p>1 - Avaliação por frequência</p> <p>Os alunos estão sujeitos às mesmas regras de avaliação.</p> <p>2 - Avaliação por Exame</p> <p>Os alunos estão sujeitos às mesmas regras de avaliação.</p>