

### Caraterização da Unidade Curricular / Characterisation of the Curricular Unit

<b>Designação da Unidade Curricular / Curricular Unit:</b>	[3182842] Programação Avançada [3182842] Advanced Programming		
<b>Plano / Plan:</b>	Licenciatura em Engenharia Eletrotécnica V2		
<b>Curso / Course:</b>	Licenciatura em Engenharia Eletrotécnica Electrical Engineering		
<b>Grau / Diploma:</b>	Licenciado		
<b>Departamento / Department:</b>	Departamento de Engenharia Electrotécnica		
<b>Unidade Orgânica / Organic Unit:</b>	Escola Superior de Tecnologia e Gestão de Viseu		
<b>Área Científica / Scientific Area:</b>	Ciências de Base		
<b>Ano Curricular / Curricular Year:</b>	1		
<b>Período / Term:</b>	S2		
<b>ECTS:</b>	4.5		
<b>Horas de Trabalho / Work Hours:</b>	0119:00		
<b>Horas de Contacto/Contact Hours:</b>			
(T) Teóricas/Theoretical:	0013:00	(TC) Trabalho de Campo/Fieldwork:	0000:00
(TP) Teórico-Práticas/Theoretical-Practical:	0000:00	(OT) Orientação Tutorial/Tutorial Orientation:	0000:00
(P) Práticas/Practical:	0026:00	(E) Estágio/Internship:	0000:00
(PL) Práticas Laboratoriais/Practical Labs:	0000:00	(O) Outras/Others:	0000:00
(S) Seminário/Seminar:	0000:00		

### Docente Responsável / Responsible Teaching

[3364] Rui Pedro De Oliveira Alves

### Docentes que lecionam / Teaching staff

[3364] RUI PEDRO DE OLIVEIRA ALVES

### **Objetivos de Aprendizagem**

Os objectivos da unidade curricular de Programação Avançada visam dotar os alunos de conhecimentos de programação orientada a objectos para a implementação de estruturas de dados e algoritmos complexos.

Os alunos aprovados a esta unidade curricular deverão adquirir as seguintes competências:

- Conhecer o paradigma orientado a objectos, reconhecendo as suas características fundamentais;
- Conceptualizar e desenhar o modelo de uma aplicação, recorrendo a diagramas de classes, de acordo com a notação UML;
- Codificar o diagrama de classes em C++;
- Elaborar estruturas de dados complexas em C++.

### **Learning Outcomes of the Curricular Unit**

This UC provides students skills in the area of object-oriented programming, for the implementation of data structures and complex algorithms.

After doing this discipline, students should be able to:

- Understand the object oriented paradigm, recognizing its fundamental characteristics;
- Conceptualize and design software design using class diagrams, according to the UML notation;
- Implement class diagrams in C++;
- Develop complex data structures in C++.

### **Conteudos Programáticos (Lim:1000)**

1. A linguagem de programação C (revisões)
2. Programação Orientada a Objectos
  - Conceitos básicos
  - Classes
  - Herança
  - Polimorfismo
3. Modelação
  - A linguagem UML
  - Diagramas de Classes
4. Programação em C++
5. Implementação de estruturas de dados em C++ com recurso à STL - Standard Template Library
  - Vector
  - Queue
  - Priority Queue

### **Syllabus (Lim:1000)**

1. Revisions on C language
2. Object Oriented Programming
  - Fundamentals
  - Classes
  - Inheritance
  - Polymorphism
3. Modelling
  - UML language
  - Class Diagrams
4. C++ Programming
5. Data structures implementation em C++ by using the STL - Standard Template Library
  - Vector
  - Queue
  - Priority Queue

### **Metodologias de Ensino (Avaliação incluída; Lim:1000)**

As aulas teóricas são dadas de acordo com o método expositivo, promovendo a interação constante com os alunos.

As aulas práticas baseiam-se na resolução de fichas práticas sobre a linguagem C++, utilizando ferramentas de desenvolvimento para esta linguagem.

Os alunos são avaliados durante as aulas, recorrendo a um método de avaliação contínua, utilizando uma plataforma online.

A avaliação da disciplina obedece aos seguintes critérios:

1. Por frequência: dois testes práticos, cada um com nota mínima de 7 valores e correspondendo a 50% da nota final.
2. Por exame: teste prático único com toda a matéria.

### **Teaching Methodologies (Including evaluation; Lim:1000)**

It is used the expository method in the theoretical classes with the support of a video-projector.

Practical classes are based on solving practical information sheets about the C++ language, using appropriate development tools.

Students are evaluated in the classes periodically, using the online platform.

Evaluation Rules:

1. Continuous evaluation: two practical works, 50% of the final grade each (min. 7 values)
2. Final Exam evaluation: 100% of the final grade

### **Bibliografia de Consulta (Lim:1000)**

- Diapositivos da disciplina
- "C++ primer", S. Lippman, Addison-Wesley, 2ª edição; COTA: 004.43 C++ LIP PRI
- "Programação Baseada em Objectos", J. Delgado, INESC
- "Programação Orientada para Objectos", Brad J. Cox Addison/Wesley - 1986
- "Programação em C++", J. Berry, MAKRON Books
- "Programação em C++", P. Rodrigues, P. Pereira, M. Sousa, FCA, 2000; COTA: 004.43 C++ ROD PRO
- "Object-oriented Software Construction", B. Meyer, Prentice-Hall, 1988; COTA: 004.43 OBJECTOS ZUL
- "Thinking in C++", B. Eckel, 2ª edição, Free Electronic Book Volume 1 e 2,  
<http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html>

### **Bibliography (Lim:1000)**

- Theoretical classes slides and practical worksheets.
- "C++ primer", S. Lippman, Addison-Wesley, 2ª edição; COTA: 004.43 C++ LIP PRI
- "Programação Baseada em Objectos", J. Delgado, INESC
- "Programação Orientada para Objectos", Brad J. Cox Addison/Wesley - 1986
- "Programação em C++", J. Berry, MAKRON Books
- "Programação em C++", P. Rodrigues, P. Pereira, M. Sousa, FCA, 2000; COTA: 004.43 C++ ROD PRO
- "Object-oriented Software Construction", B. Meyer, Prentice-Hall, 1988; COTA: 004.43 OBJECTOS ZUL
- "Thinking in C++", B. Eckel, 2ª edição, Free Electronic Book Volume 1 e 2,  
<http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html>

### **Observações**

«Observações»

### **Observations**

«Observations»

Observações complementares

Inválido para efeito de certificação