

Unidade Curricular	Programação Orientada por Objetos		Área Científica	Ciências da Computação	
Licenciatura em	Engenharia Informática		Escola	Escola Superior de Tecnologia e Gestão de Bragança	
Ano Letivo	2019/2020	Ano Curricular	2	Nível	1-2
Tipo	Semestral	Semestre	1	Créditos ECTS	6.0
Código			9119-606-2104-00-19		
Horas totais de trabalho	162	Horas de Contacto	T -	TP 60	PL -
			TC -	S -	E -
			OT -	O -	

T - Ensino Teórico; TP - Teórico Prático; PL - Prático e Laboratorial; TC - Trabalho de Campo; S - Seminário; E - Estágio; OT - Orientação Tutoria; O - Outra

Nome(s) do(s) docente(s) Paulo Duarte Ferreira Gouveia, Jose Paulo Machado Da Costa

### Resultados da aprendizagem e competências

No fim da unidade curricular o aluno deve ser capaz de:

1. Identificar os princípios que regem a programação orientada por objetos
2. Construir diagramas de classes e de comunicação UML e implementar soluções com base na descrição de problemas
3. Definir classes, objetos, atributos e métodos com a linguagem C++, identificando e definindo os construtores necessários para a correta inicialização das instâncias
4. Reconhecer a necessidade de implementação de construtores de cópia, operadores de atribuição e destrutores como forma de manipular classes com atributos dinâmicos
5. Distinguir a agregação das associações simples e realizar convenientemente a sua implementação em C++
6. Identificar e implementar a herança entre classes e estabelecer hierarquias de classes
7. Compreender o conceito de polimorfismo e definir e aplicar funções virtuais como forma de o implementar
8. Compreender os conceitos de classe abstrata e funções virtuais puras como forma de impor funcionalidades nas classes derivadas

### Pré-requisitos

Antes da unidade curricular o aluno deve ser capaz de:

1. Ser capaz de efetuar raciocínios lógicos para a resolução de problemas
2. Criar programas utilizando o paradigma procedimental

### Conteúdo da unidade curricular

Definição de Programação Orientada por Objetos. Princípios da Programação Orientada por Objetos. Conceitos de Modelação Orientada por Objetos. Introdução à linguagem C++. Definição de classes em C++. Implementação de associações. Templates de funções e de classes. Bibliotecas standard do C++. Implementação da herança e de hierarquias de classes. Gestão de memória dinâmica interna a uma classe. Coleções híbridas. Entrada e saída de dados. Manipulação de ficheiros.

### Conteúdo da unidade curricular (versão detalhada)

1. Definição da Programação Orientada por Objetos
  - Motivação
  - Conceitos básicos
2. Princípios da Programação Orientada por Objetos
  - Encapsulamento
  - Herança
  - Polimorfismo
3. Conceitos de Modelação Orientada para Objetos
  - Diagramas de classes em UML
  - Diagramas de comunicação em UML
  - Associações entre classes: simples, agregação e composição
  - Herança: sobreposição e acréscimo de características, classes abstratas, herança múltipla
4. Introdução à linguagem C++
  - MS Visual Studio
  - Declarações
  - Constantes
  - Tipos de dados
  - Expressões e operadores
  - Funções
5. Definição de classes em C++
  - Atributos
  - Construtores. Categorias de construtores
  - Métodos
6. Funcionalidades básicas do C++
  - Arrays e apontadores de objetos. Autorreferência nas classes
  - Membros constantes
  - Sobre carga de operadores (Overloading)
7. Implementação de associações simples e agregação de classes
8. Templates de funções e de classes
9. Bibliotecas standard do C++
  - Classes string e set
  - O template de classes 'Colecao'
10. Implementação de associações
  - Associações e coleções
  - Coleções de cópia e referência
  - Associações 1-N
  - Associações N-N
  - Classes associativas
11. Referências
  - Definição de referências
  - Passagem de parâmetros e retorno
12. Implementação da herança e de hierarquias de classes
  - Acréscimo e substituição de métodos
  - Construtores e herança. Listas de inicialização
  - Tipos de proteção no acesso aos membros
  - Conversão ascendente (Upcast) e descendente (downcast)
  - Polimorfismo e funções virtuais
  - Classes abstratas e funções virtuais puras
13. Gestão de memória dinâmica interna a uma classe
  - Construtor de cópia
  - Destrutor

**Conteúdo da unidade curricular (versão detalhada)**

- Operador afetação
- 14. Agregação com apontadores
- 15. Implementação de coleções híbridas
- 16. Operadores de conversão. Membros estáticos
- 17. Declarações 'friend'
- 18. Entrada e saída de dados e manipulação de ficheiros

**Bibliografia recomendada**

1. The C++ Programming Language (4th Edition). Bjarne Stroustrup, Addison-Wesley Professional, 2013
2. Thinking in C++, Volume 1 (2nd Edition). Bruce Eckel, Prentice Hall, 2000
3. Object-Oriented Modeling and Design with UML (2nd Edition). Michael R. Blaha, James R. Rumbaugh, Prentice Hall, 2004

**Métodos de ensino e de aprendizagem**

Exposição teórica de conceitos acompanhada pela discussão prática de exemplos modelo. Aplicação dos conceitos através da resolução de pequenos exercícios práticos que exemplifiquem a sua utilização. Resolução de um problema que permita a aplicação global de todos os conceitos adquiridos. Execução de um projeto final.

**Alternativas de avaliação**

1. Alternativa 1 - (Ordinário, Trabalhador) (Final)
  - Prova Intercalar Escrita - 30%
  - Trabalhos Práticos - 20%
  - Exame Final Escrito - 50% (Componente com nota mínima de 7 valores em 20.)
2. Alternativa 2 - (Ordinário, Trabalhador) (Recurso, Especial)
  - Exame Final Escrito - 100% (Inclui exercícios suplementares destinados a substituir a Prova Intercalar e o Trabalho Prático.)

**Língua em que é ministrada**

1. Português
2. Inglês

**Validação Eletrónica**

Paulo Duarte Ferreira Gouveia	José Luís Padrão Exposto	José Carlos Rufino Amaro	Paulo Alexandre Vara Alves
31-10-2019	12-11-2019	12-11-2019	13-11-2019