INSTITUTO POLITÉCNICO DE BRAGANÇA

Unidade Curricular	Programação Orientada por Objetos	Área Científica	Ciências da Computação
Licenciatura em	Engenharia Informática	Escola	Escola Superior de Tecnologia e de Gestão de Bragança
Ano Letivo	2025/2026	Ano Curricular	2
Nível	1-2	Créditos ECTS	6.0
Tipo	Semestral	Semestre	1
Código	9119-850-2103-00-25		

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 Tutórica; O - Outra

Nome(s) do(s) docente(s)	Paulo Duarte Ferreira Gouveia, Carlos Eduardo Castro Correia, Daniel Jose Lopes Gouveia, Gabriel de Arruda Leite, Helder Francisco Silva Vieira, Jose Paulo Machado Da Costa

Resultados da aprendizagem e competências

No fim da unidade curricular o aluno deve ser capaz de:
1. Compreender e aplicar os princípios fundamentais da Programação Orientada por Objetos (POO), incluindo encapsulamento, herança e polimorfismo.
2. Criar e interpretar diagramas de classes em UML, para modelar sistemas baseados no paradigma orientado a objetos.
3. Projetar e implementar classes e objetos em C++, definindo os atributos e métodos adequados, bem como garantindo 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. Diferenciar e implementar associações simples e agregação entre classes, aplicando estruturas de dados adequadas para colecionamento dinâmico.
6. Estabelecer hierarquias de classes e aplicar corretamente os conceitos de herança em C++, promovendo a reutilização de código.
7. Implementar polimorfismo através de funções virtuais, permitindo a extensão e adaptação de comportamentos em classes derivadas.
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. Aplicar raciocínio lógico na análise e conceção de soluções computacionais.
2. Desenvolver programas no paradigma procedimental, utilizando estruturas fundamentais de programação.

Conteúdo da unidade curricular

Conceitos, motivação e princípios fundamentais da Programação Orientada por Objetos. Modelação com diagramas de classes UML. 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 e acesso a 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 por 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
   - Sobrecarga 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

Este documento só tem validade académica depois de autenticado, em todas as suas folhas, com o selo a óleo da Instituição.

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 acesso a ficheiros

Bibliografia recomendada

1. The C++ Programming Language (4th Edition). Bjarne Stroustrup, Addison-Wesley, 2013
2. C++ Programming: An Object-Oriented Approach. B. Behrouz A. Forouzan and Richard F. Gilberg, McGraw-Hill Education, 2020
3. Fundamentals of C++ Programming. Richard L. Halterman, School of Comp. South. Adv. University US, 2018
4. C++: Guia Moderno de Programação. Henrique Loureiro, FCA – Editora de Informática, 2019
5. Modelação de Dados em UML – uma abordagem por problemas. Borges, T. Dias e J. Cunha, FCA – Editora de Informática, 2015

Métodos de ensino e de aprendizagem

A exposição teórica de conceitos é complementada pela discussão e análise de exemplos práticos, permitindo a compreensão gradual dos fundamentos da POO. A aprendizagem é reforçada através da resolução de exercícios progressivos e de um projeto final, no qual se implementa uma solução completa em C++, promovendo a integração dos conhecimentos adquiridos ao longo da unidade curricular.

Alternativas de avaliação

1. Alternativa 1 - (Ordinário, Trabalhador) (Final)
   - Trabalhos Práticos - 50% (Modelação e implementação em C++ de uma solução no paradigma POO)
   - Exame Final Escrito - 50% (Componente com nota mínima de 7.0 valores em 20.)
2. Alternativa 2 - (Ordinário, Trabalhador) (Recurso, Especial)
   - Exame Final Escrito - 100%

Língua em que é ministrada

1. Português
2. Inglês

Validação Eletrónica

Paulo Duarte Ferreira Gouveia	Tiago Miguel Ferreira Guimaraes Pedrosa	Isabel Maria Lopes	José Carlos Rufino Amaro
14-10-2025	15-10-2025	15-10-2025	01-11-2025

Este documento só tem validade académica depois de autenticado, em todas as suas folhas, com o selo a óleo da Instituição.