

Course Unit	Object Oriented Programming		Field of study	Computer Science	
Bachelor in	Informatics Engineering		School	School of Technology and Management	
Academic Year	2021/2022	Year of study	1	Level	1-1
Type	Semestral	Semester	2	ECTS credits	6.0
Workload (hours)			162	Contact hours	
			T	-	TP
			60	PL	-
			TC	-	S
			E	-	OT
			O	-	
Code 9119-706-1204-00-21					

T - Lectures; TP - Lectures and problem-solving; PL - Problem-solving, project or laboratory; TC - Fieldwork; S - Seminar; E - Placement; OT - Tutorial; O - Other

Name(s) of lecturer(s) Paulo Duarte Ferreira Gouveia, Jose Paulo Machado Da Costa, Nelson Alexandre Perdigao Figueiredo, Nuno Romeu Cardoso Sequeira

### Learning outcomes and competences

At the end of the course unit the learner is expected to be able to:

1. Identify the guiding principles of object oriented programming
2. Create UML class and communication diagrams and implement solutions based on problems descriptions
3. Define classes, objects, attributes and method using C++ language, identifying and defining the needed constructors to the correct instance initialization
4. Recognize the need to implement copy constructors, assignment operators and destructors to handle dynamic class attributes
5. Distinguish aggregation and simple associations and accomplish their implementation conveniently
6. Identify and implement inheritance between classes and establish class hierarchies
7. Understand the concept of polymorphism and implement it by means of the definition and application of virtual functions
8. Understand the concept of abstract class and pure virtual functions as a mean to impose operations in the derived classes

### Prerequisites

Before the course unit the learner is expected to be able to:

1. Elaborate logical reasoning for the resolution of problems
2. Create programs using the procedural paradigm

### Course contents

Object Oriented Programming definition. Principles of Object Oriented Programming. Concepts of object oriented modeling. Introduction to the C++ language. Class definition in C++. Implementation of associations. Class and functions templates. C++ Standard libraries. Implementation of inheritance and class hierarchies. Dynamic memory management inside a class. Hybrid collections. Input and output and file handling.

### Course contents (extended version)

1. Object Oriented Programming definition
  - Motivation
  - Basic concepts
2. Principles of Object Oriented Programming
  - Encapsulation
  - Inheritance
  - Polymorphism
3. Concepts of object oriented modeling
  - UML class diagrams
  - UML communication diagrams
  - Associations between classes: simple, aggregation and composition
  - Inheritance: overriding and adding new features, abstract classes, multiple inheritance
4. Introduction to the C++ language
  - MS Visual Studio
  - Declarations
  - Constants
  - Data types
  - Expressions and operators
  - Functions
5. Class definition in C++
  - Attributes
  - Constructors. Constructors categories
  - Methods
6. Basic features of C++
  - Object arrays and object pointers. Self reference
  - Constant members
  - Operator overloading
7. Implementation of simple associations and aggregation
8. Function and class templates
9. C++ Standard libraries
  - String and set classes
  - Collection class template
10. Implementation of associations
  - Associations and collections
  - Copy collections and reference collections
  - 1-N associations
  - N-N associations
  - Associative classes
11. References
  - Reference definition
  - Parameter passage and return
12. Implementation of inheritance and class hierarchies
  - Method addition and overriding
  - Constructors and inheritance. Initializer list
  - Types of member access protection
  - Upcast and downcast
  - Polymorphism and virtual functions
  - Abstract classes and pure virtual functions
13. Dynamic memory management inside a class
  - Copy constructor
  - Destructor
  - Assignment operator
14. Aggregation with pointers

**Course contents (extended version)**

15. Implementation of hybrid collection
16. Conversion operators. Static members
17. Friend declarations
18. Input and output and file handling

**Recommended reading**

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

**Teaching and learning methods**

Lecture classes of theoretical concepts followed by practical discussion of model examples. Concept application through small problem solving. Practical experience is developed with the resolution of a larger problem. Execution of a final project assignment.

**Assessment methods**

1. Alternative 1 - (Regular, Student Worker) (Final)
  - Practical Work - 50% (Modeling and implementation in C++ of a solution in the OOP paradigm)
  - Final Written Exam - 50% (Component with a minimum score of 7 out of 20.)
2. Alternative 2 - (Regular, Student Worker) (Supplementary, Special)
  - Final Written Exam - 100%

**Language of instruction**

Portuguese

**Electronic validation**

Paulo Duarte Ferreira Gouveia	José Luís Padrão Exposto	Luísa Maria Garcia Jorge	Paulo Alexandre Vara Alves
08-03-2022	12-03-2022	22-03-2022	25-03-2022