

Instituto Superior de Engenharia

60024233 - Programação Orientada a Objetos

Informação Geral

Ano Letivo	2022/2023
Semestre	1.º Semestre
Docente Responsável	João António Pereira Almeida Durães
Língua de Ensino	Português
Créditos ECTS	5
Cursos	Licenciatura em Engenharia Informática
Modo de Ensino	Presencial
Outros Idiomas Inf. Edição	Inglês
Informação copiada da ficha de edição do ano anterior sem alterações	Não
Data de Submissão	23-09-2022
Data de Apreciação	
Estado da Proposta	Por Apreciar

Objectivos e Competências a Desenvolver

Português

1. Adquirir competências de programação orientada a objectos.
2. Adquirir competências de programação orientada a objectos em C++.
3. Adquirir a capacidade de desenvolver aplicações para ambiente consola em C++.
4. Adquirir a capacidade de desenvolver aplicações baseadas em estruturas de dados encapsuladas e polimórficas.
5. Aperfeiçoar a capacidade de organização, pesquisa, planeamento, justificação de soluções e documentação de projetos de software.
6. Aperfeiçoar a capacidade de pesquisa autónoma de informação.

Objectivos e Competências a Desenvolver

Inglês

1. Acquire object-oriented programming skills.
2. Acquire object-oriented programming skills in C++.
3. Acquire the ability to develop console C++ applications.
4. Acquire the ability to develop applications based on encapsulated and polymorphic data structures.
5. Improve the ability to organize, research, plan, justify solutions and document software projects.
6. Improve the aptitude for autonomous information research.

Programa

Português

Introdução geral aos conceitos de orientação a objetos e à linguagem C++

Conceitos gerais sobre Programação Orientada ao Objecto

Encapsulamento, Polimorfismo, Herança

Princípios SOLID

Algumas diferenças entre C e C++

Entrada e saída de dados standard

Estruturação de projetos em .h e .cpp

Comentários em C++

Definição de variáveis locais

Tipos pré-definidos

Cast

Referências

Constantes

Funções

Funções overloaded

Funções com argumentos por omissão

Situações de ambiguidade em overloading

Classes

Sintaxe de definição de classes

Classes como evolução de estruturas

Funções membros

O ponteiro this

Funções inline

Visibilidade (acessibilidade)

Construtores

Sintaxe de inicialização () e {}

initializer_list

Arrays não-dinâmicos de objetos

Destrutores

Funções membros inline

dados e funções membros estáticos

Funções membros constantes

Funções friend

Ciclo de vida dos objetos

Destrutor

Introdução à construção por cópia

atribuição. Introdução ao operador de atribuição

Objetos dinâmicos

Os operadores new e delete

Arrays dinâmicos de objetos

Redefinição de operadores

Introdução geral à redefinição de operadores

Funções operadores

Operadores unários e binários

Opção de operadores membro vs. operadores não membro

Conversões

Composição e agregação

Significado de composição por oposição a agregação

Composição sem memória dinâmica - sintaxe, requisitos de inicialização. Ordem de chamada

constructores e de destrutores

Agregação por referência e por ponteiro

Composição através de memória dinâmica

Conceito de gestão de recursos próprios em classes

Membros variáveis que são ponteiros para memória dinâmica

Destrutor, Operador atribuição, Construtor por cópia

Composição de objectos

Composição de objectos e constructores

Tipos básicos na lista de inicialização

Composição de objectos e destrutores

Herança em C++

Classes derivadas

Composição vs. derivação

Formas de derivação e acessibilidade

Constructores e destrutores

Construtor por cópia e operador atribuição em classes derivadas

Herança e funções overloaded

Polimorfismo

Upcasting

Funções virtuais

Classes abstractas e funções virtuais puras

Funções virtuais overloaded e redefinição em classes derivadas

Destrutores virtuais

Mecanismo de excepções em C++

Introdução à biblioteca STL

Noção de contentor e de iterador

Classes de contentores. Manipulação de contentores. Classe vector.

classes de suporte a manipulação de strings. string, e ostream.

Classes para suporte a E/S por stream istream e ostream

Introdução a smartpointers em C++

unique pointer

shared pointer

Introdução aos templates e biblioteca STL

Templates de classes

Argumentos de templates

Templates de funções

Programa

Inglês

General introduction to object-oriented concepts and the C++ language

- General Concepts of Object Oriented Programming
- Encapsulation, Polymorphism, Inheritance
- SOLID Principles

Some differences between C and C++

- Standard data input and output Structuring projects in .h and .cpp
- Comments in C++
- Definition of local variables
- Predefined types
- Type cast
- References
- Constants
- Functions
- Overloaded Functions
- Functions with default arguments
- Ambiguity situations in overloading

Classes

- Class definition syntax
- Classes as Structure evolution
- Member functions

- The this pointer
- Inline Functions
- Visibility (accessibility)
- Constructors
- Initialization syntax () and {}
- initializer_list
- Non-dynamic arrays of objects
- Destructors
- Inline member functions
- static members: data and functions
- constant member functions
- friend functions

Objects life cycle

- Destructor
- Introduction to copy construction
- Assignment. Introduction to the assignment operator

Dynamic objects

- The new and delete operators
- Dynamic arrays of objects

Operator overloading

- General Introduction to Operator Overloading
- Operator Functions
- Unary and binary operators
- Choice of member operators vs. non-member operators
- Conversions operators

Composition and aggregation

- Meaning of composition versus aggregation

- Composition without dynamic memory - syntax, initialization requirements. Constructors and Destructors calling order
- Aggregation by reference and pointer

Composition through dynamic memory

- Concept of managing own resources in classes
- Data members that are pointers to dynamic memory
- Destructor, Assignment Operator, Copy Constructor

Object composition

- Composition of objects and constructors
- Basic Types in initialization list
- Composition of objects and destructors

Inheritance in C++

- Derived classes
- Composition vs. derivation
- Forms of derivation and accessibility
- Constructors and destructors
- Copy constructor and assignment operator on derived classes
- Inheritance and overloaded functions

Polymorphism

- Upcasting
- Virtual functions
- Abstract classes and pure virtual functions
- Overloaded virtual functions and redefinition in derived classes
- Virtual destructors

C++ Exception Mechanism

Introduction to the STL library

- Notion of container and iterator
- Classes of containers. Container manipulation.
- Class vector. Class map. Classes for string manipulation. String, istringstream, ostringstream
- Classes for stream I/O support ifstream and ofstream

Introduction to smartpointers in C++

- unique pointer
- shared pointer
- weak pointer

Introduction to templates and STL library

- Class Templates
- Template arguments
- Function templates

Conhecimentos Base Recomendados

Português

Programação em linguagem C e algoritmia. É importante os alunos terem obtido aprovação nas unidades curriculares de Introdução à Programação e Programação do 1º ano.

Conhecimentos Base Recomendados

Inglês

Programming in C language and algorithms. It is important that students have passed the 1st year Introduction to Programming and Programming curricular units.

Métodos de Ensino

Português

As aulas são teóricas e práticas. Nas aulas teóricas, é utilizada uma metodologia expositiva para a apresentação dos fundamentos teóricos da matéria com exemplos de aplicação. Nas aulas práticas os alunos aplicam os conhecimentos das aulas teóricas, desenvolvendo aplicações, com a supervisão dos docentes. Fora das aulas, os alunos desenvolvem, em grupos de dois, um projeto de média dimensão que é avaliado

numa fase intermédia e numa fase final. Os docentes acompanham o desenvolvimento do projeto, esclarecendo dúvidas e orientando quanto às opções tomadas.

Métodos de Ensino

Inglês

The classes are theoretical and practical. In theoretical classes, an expository methodology is used to present the theoretical foundations of the subject with application examples. In the practical classes, students apply the knowledge of theoretical classes, developing applications, under the supervision of teachers. Outside the classes, the students develop, in groups of two, a medium-sized project that is evaluated in an intermediate phase and in a final phase. The teachers monitor the development of the project, clearing up doubts and providing guidance on the options taken.

Métodos de Avaliação

Português

Em cada uma das épocas associadas a este ano lectivo, a classificação final do aluno resulta da média ponderada de uma componente escrita e de uma componente prática, com os pesos de 60% (12 valores em 20) e de 40% (8 valores em 20), respetivamente.

Componente escrita - exame

- A prova escrita (exame) é sem consulta.
- É obrigatória a obtenção de uma classificação igual ou superior a 35% no exame. A não obtenção desta classificação mínima implica a reprovação à disciplina.

Componente prática – projeto / trabalho prático

- A componente prática resulta da classificação obtida num trabalho prático entregue em duas fases.
- O trabalho prático consiste no desenvolvimento de um problema proposto pelos docentes, e é realizado em grupos de dois alunos.
- O trabalho é sujeito a defesa individual nas duas fases sem a qual não será atribuída nota.
- São aceites trabalhos efetivamente realizados no ano lectivo 2021/2022 desde que tenham obtido uma classificação superior ou igual a 50%.

As duas fases do trabalho são regidas da seguinte forma:

1ª Fase

- A entrega da primeira fase ocorre no meio do semestre, em data especificada no enunciado. Data prevista: 19 Nov., 2022
- Na primeira fase é entregue uma versão preliminar do trabalho, incidindo principalmente na interface com o utilizador e aspectos que não exigem ainda conhecimento aprofundado das características mais avançadas de C++. Esta meta serve para promover o trabalho atempado pelos

alunos.

- A qualidade do material entregue na primeira fase traduz-se por um número entre 0.8 e 1.0 pelo qual será multiplicado a nota obtida globalmente pelo trabalho, entregue na fase 2.
- O valor obtido na meta 1 não sem a meta 2 não serve de classificação.

2ª Fase

- A entrega da segunda fase ocorre no final das aulas do semestre, em data especificada no enunciado. Data prevista: 7 Jan., 2023
- Na segunda fase é entregue o trabalho na sua totalidade.
- Nesta fase o trabalho é avaliado na globalidade e, juntamente com a qualidade da defesa, chega-se ao valor de avaliação que é então multiplicado pelo factor entre 0.8 e 1.0 que foi determinado pela qualidade do material entregue na primeira fase. O valor resultante é a nota do trabalho.

Regras gerais do trabalho prático

- Não são aceites grupos de 3 ou mais alunos.
- A defesa do trabalho prático é obrigatória nas duas fases de entrega.
- Os alunos do mesmo grupo podem ter nota diferentes consoante a prestação na defesa.
- O trabalho não tem nota mínima.
- A nota do trabalho prático feito ao longo do semestre será considerada em todas as épocas de exame a realizar durante o ano letivo (normal, recurso e especial).
- O trabalho é realizado durante o semestre e entregue antes dos exames. Não existem datas de entrega alternativas noutras épocas. Não existe a possibilidade de novas entregas na época especial ou outra qualquer para além da época normal.

Métodos de Avaliação

Inglês

In each of the exam calls associated with this academic year, the student's final grade results from the weighted average of a written component and a practical component, with the weights of 60% (12 points out of 20) and 40% (8 points out of 20), respectively.

Written component – exam

- The written test (exam) is without consultation.
- It is mandatory to obtain a grade equal to or greater than 35% on the exam. Failure to obtain this minimum classification implies disapproval of the subject.

Practical component - project / practical work

- The practical component results from the classification obtained in a practical work delivered in two phases.

- The practical work consists of developing a problem proposed by the teachers, and is carried out in groups of two students.
- The work is subject to individual defense in the two phases, without which no grade will be awarded.
- Works developed in the academic year 2021/2022 are accepted as long as they have obtained a grade greater than or equal to 50%.

The two phases of the work are governed as follows:

1st Phase

- The delivery of the first phase takes place in the middle of the semester, on a date specified in the statement. Expected date: 19 Nov, 2022
- In the first phase, a preliminary version of the work is delivered, focusing mainly on the user interface and aspects that do not yet require in-depth knowledge of the most advanced features of C++. This goal serves to promote timely work by students.
- The quality of the material delivered in the first phase is reflected in a number between 0.8 and 1.0 by which the grade obtained globally by the work, delivered in phase 2 will be multiplied.
- The value obtained in goal 1 not without goal 2 is not a classification.

2nd Phase

- Delivery of the second phase takes place in the middle of the semester, on a date specified in the statement. Expected date: Jan 7, 2023
- In the second phase, the entire work is delivered.
- In this phase, the work is evaluated as a whole and, together with the quality of the defense, the assessment value is arrived at, which is then multiplied by the factor between 0.8 and 1.0, which was determined by the quality of the material delivered in the first phase. The resulting value is the grade for the assignment.

General rules of practical work

- Groups of 3 or more students are not accepted.
- The defense of the practical work is mandatory in the two delivery phases.
- Students from the same group may have different grades depending on their defense performance.
- The work has no minimum grade.
- The grade of the practical work done during the semester will be considered in all exam periods to be carried out during the academic year (normal, recourse and special).
- The work is carried out during the semester and delivered before exams. There are no alternative delivery dates at other times. There is no possibility of new deliveries in the special exam call or any other than the first one "época normal".

Bibliografia

Português

Referências principais

Brokken, F. B. (2021). C++ Annotations Version 12.2.0, University of Groningen publishing, <http://www.icce.rug.nl/documents/cplusplus/> - Online

Stroustrup, B. (2000). A linguagem de programação C++, Bookman – 1ª-1-359

Referências complementares

Lippman, S.B., Lajoie, J., Moo B. E. (2013). C++ Primer (5th ed.), Addison-Wesley

Eckel, B. (2000). Thinking in C++ Vol.1 (2nd ed.), Prentice Hall Inc., <http://www.bruceeckel.com/> - 1A-1-380 / Online

Deitel, P. J., Deitel, H. M. (2010). C++ How to Program (7th edition), Prentice Hall Inc. – 1A-1-425

Bibliografia

Inglês

Main references

Brokken, F. B. (2021). C++ Annotations Version 12.2.0, University of Groningen publishing, <http://www.icce.rug.nl/documents/cplusplus/> - online

Stroustrup, B. (2000). A linguagem de programação C++, Bookman – 1ª-1-359

Auxiliary references

Lippman, S.B., Lajoie, J., Moo B. E. (2013). C++ Primer (5th ed.), Addison-Wesley

Eckel, B. (2000). Thinking in C++ Vol.1 (2nd ed.), Prentice Hall Inc., <http://www.bruceeckel.com/> - 1A-1-380 / Online

Deitel, P. J., Deitel, H. M. (2010). C++ How to Program (7th edition), Prentice Hall Inc. – 1A-1-425

Atividades de promoção da ligação do estudante a atividades de desenvolvimento da investigação, transferência de tecnologia e conhecimento e inovação

Português

Poderão vir a ser propostos trabalhos individuais de carácter avançado com interesse para a matéria mediante interesse demonstrado pelo aluno, com valorização própria mediante o plano do trabalho. Esta situação constitui uma situação de carácter excecional, individualizada, que poderá ocorrer apenas com o acordo com os docentes, mediante valor e interesse demonstrados pelo aluno envolvido, e dentro das

possibilidades operacionais da unidade curricular.

Atividades de promoção da ligação do estudante a atividades de desenvolvimento da investigação, transferência de tecnologia e conhecimento e inovação

Inglês

Individual works of an advanced nature may be proposed with an interest in the subject based on the interest shown by the student, with their own appreciation through the work plan. This situation constitutes an exceptional, individualized situation, which can only occur with the agreement of the teachers, according to the value and interest shown by the student involved, and within the operational possibilities of the curricular unit.