



**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE RESERVATÓRIO E DE EXPLORAÇÃO
(MESTRADO E DOUTORADO ACADÊMICO - STRICTO SENSU)**

PROGRAMA ANALÍTICO DE DISCIPLINA (EMENTA)

Sigla: **LEP1732** Nome: **Desenvolvimento de software orientado a objeto em C++**

Horas teórica: **51** Horas prática: **34** Horas extraclasse: 0 Carga horária total: **85**
Créditos: 4 Tipo de aprovação: **Média/Frequência**

Ementa:

Introdução ao C++; História, Modern C++/C++2; Etapas de compilação; Diretrizes de pré-processador; Módulos; Conceitos básicos de C++, sintaxe, escopo; Operadores; Estruturas de controle e repetição; Referências; Funções; Sobrecarga de funções; Tipos (padrões, do usuário, de bibliotecas); Gerenciamento de memória; Estruturas de dados; Funções lambda; Orientação a Objetos com C++; classes, objetos, atributos; Métodos e sobrecarga de métodos; Construtor e destrutor, Associações, amizade/friend, herança, [herança múltipla], polimorfismo; Sobrecarga e de operadores; A biblioteca de entrada e saída de dados (streams), redirecionamento, acesso a arquivos de disco (filesystem); A biblioteca de gabaritos de C++ (a STL), iteradores, containers (<<vector>>, <<deque>>, <<list>>, <<stack>>, <<queue>>, <<set>>, <<map>>), objetos funções e funções genéricas; As bibliotecas numéricas de C++: cmath, numbers, <<random>>, <<complex>>, <<random>>; Bibliotecas úteis: <<pair>>, <<tuple>>, <<variant>>, <<optional>>, <<any>>. Ensino por metodologia tradicional (teoria/aplicação prática) ou através do uso de projetos de engenharia. Nota: As ferramentas a serem utilizadas podem mudar, os nomes são apenas indicativos/ilustrativos. Site Professor: <https://sites.google.com/view/professorandreuartebueno/>

Conteúdo programático:

1. Atividades Teóricas:
2. Apresentação da ementa e metodologia da disciplina.
3. Introdução ao C++; Porque C++; História; Modern C++/C++2; Ansi ISO C++;
4. Etapas de compilação; Tempo de compilação e execução; Diretrizes de pré-processador; Módulos; Namespace; *static_assert*;
5. Conceitos básicos de C++, palavras chave, declaração, definição, sintaxe, escopo, operadores;
6. Estruturas de controle e repetição;
7. Referências; Funções; Sobrecarga de funções; Funções lambda; ***std::function***;
8. Apresenta a programação orientada a objeto em C++; Quais as características de um programa orientado a objeto usando C++;
9. Introdução ao conceito de tipos: o que é um tipo. O uso dos tipos predefinidos de C++, incluindo o consumo de memória e valores limites dos tipos predefinidos, exemplos de uso de caracteres, números inteiros, flutuantes e unsigned. O uso de **auto**. Usando sufixos para os tipos predefinidos. A definição de tipos de usuário, como vetores, estruturas e enumerações. A classe de strings padrões de C++ a <string>. A classe <vector>. Os tipos definidos em bibliotecas externas. Introdução aos gabaritos - template, e a biblioteca STL. Como manipular os tipos de C++, typedef, size_t(), sizeof(), typeid(), **constexpr**, **constexpr**, **constexpr**. Vantagem da tipificação forte.
10. Gerenciamento de memória; Ponteiros, **nullptr**, **smart pointer (shared_ptr, unique_ptr)**, [rvalue reference];
11. **Classes** em C++: Protótipo, Controle de acesso, Escopo.
Atributos normais, estáticos e constantes.
Métodos normais, constantes, estáticos e inline. Sobrecarga de métodos.
Como declarar, definir e usar; classes, objetos, atributos e métodos (**delete**, **default**, **override**, **final**);
12. Construtores e destrutor
13. Como implementar associações, **herança** simples, [a herança múltipla], o uso do **polimorfismo**;
14. O uso de friend, a **sobrecarga de operadores**; A classe <<complex>>
15. A biblioteca-padrão para entrada e saída de dados (a classe ios_base), os manipuladores de <iomanip>, a classe <ios>, a classe <ostream>, a classe <istream> e as classes istream e ostream localizadas em <sstream>. O uso de **format** e **print**. Como criar e usar objetos da classe <fstream> para acessar arquivos de disco. O uso de redirecionamento de entrada e saída. O acesso a dispositivos

externos como a impressora e a saída auxiliar, a criação e uso de arquivos de disco binários. Como executar e enviar comandos para outros programas usando as classes de `<fstream.h>`. Como manipular sistemas de arquivos com Filesystem. Classes para `path`, diretórios, iteradores, pipes, etc.

16. Apresenta-se a Standard Template Library (STL), que é uma biblioteca de objetos em C++; Descreve-se os conceitos básicos de containers e iteradores; Uso de `<vector>` para vetores, `<list>` para listas duplamente encadeadas, `initializer_list`, `<queue>` que representa uma fila, `<stack>` que representa uma pilha (como em uma calculadora HP), uma `<deque>` que é uma fila com duas extremidades e classes para tratamento de conjunto de dados (`<pair>`, `<tuple>`, `<set>`, `<multi_set>`, `<map>`, `<multi_map>`, `<variant>`, `<<optional>>`, `<<any>>`); Uso de funções genéricas;
17. Classes matemáticas para tratar números complexos, a classe `<complex>`, a classe `<bitset>`; Uso de classes e funções utilitárias como `<chrono>`, `<<random>>`, `<<regex>>`.
18. Opcional: Os tipos genéricos (templates); Concepts, exceções, coroutines, expressões regulares; Noções de processamento paralelo (threads) e de cluster de computadores; Apresentação de classes e exemplo básico de programação com interface gráfica usando Qt, inclui entrada e saída e uso de diálogos e gráficos - `QCustomPlot`.
19. Atividades Práticas:
20. Atividades de Extensão:
21. Atividades Extra-Classe: Exercícios a serem realizados na sala de aula/informática do LDSC e/ou em casa.

Todos os projetos desenvolvidos terão licença GPL e serão disponibilizados aos demais alunos e a comunidade através do github do LDSC (<https://github.com/ldsc>).

Bibliografia:

Como são muitas as possibilidades de projetos a serem desenvolvidos a bibliografia é extensa; A cada projeto o professor poderá orientar quais são as bibliografias centrais.

• -----Engenharia de software, especificação, elaboração, análise, projeto, teste, debugagem;

1. Sommerville, Ian; Engenharia de Software; Pearson Universidades; 10ª edição; 2019;
2. Robert C; Myers ; Essential Test-Driven Development; Addison-Wesley Professional; 2.ed; 2023;
3. André Koscianski; Qualidade de software; Novatec; 2006;
4. Leonardo Molinari; Teste de software; Érica; 2003;
5. Michal Young; Teste e análise de software; Bookman; 2008;
6. Trayahur; M; F; e Emerson Rios; Projeto e Engenharia de Software: Teste de Software; Alta Books; 2003;
7. Ann R; Ford; Practical debugging in C++; Prentice hall; 2002;

-----UML

1. Martin Fowler; UML Distilled: A Brief Guide to the Standard Object Modeling Language; Addison-Wesley Object Technology Series); 3rd Edição; 2018

-----C++

1. Stroustrup; Programming principles and practice using C++; 2 edition; 2014;
2. Bjarne, Stroustrup; C++ The Programming Language; Addison-Wesley, 4 edition; 2013;
3. Siddhartha Rao; C++ in One Hour a Day, Sams Teach Yourself; Sams Publishing; 9 edition; 2022;
4. Josuttis, N; M; Object Oriented Programming in C++; John Wiley & Sons; 2002;
5. Tony Gaddis; Starting Out with C++: From Control Structures through Objects; Addison Wesley; 1 edition revisada; 2015;
6. Malik; C++ Programming: From Problem Analysis to Program Design; 8 Edition (Paperback D;S; (Course Technology); 2017; ISBN-13 : 978-1337102087
7. John Paul Mueller; C++ All-In-One for Dummies; For Dummies; 4th ed. edição; 2021;
8. Nicolai M Josuttis; C++17 - The Complete Guide; Editora Nicojosuttis; 2019;
9. Kate Gregory; Beautiful C++: 30 Core Guidelines for Writing Clean, Safe, and Fast Code; Addison-Wesley Professional; 1ª edição; 2021;
10. Klaus Iglberger; C++ Software Design: Design Principles and Patterns for High-Quality Software; 1st Edition; O'Reilly Media; 1st edition; ISBN-13: 978-1098113162
11. Christopher Kormanyos ; Real-Time C++: Efficient Object-Oriented and Template Microcontroller Programming; Springer; 4th ed; 2021 Edition; ISBN-13: 978-3662629987
12. Paul Deitel and Harvey Deitel; C++20 for Programmers: An Objects-Natural Approach (Deitel Developer

Series); Pearson; 3rd Edition; ISBN-13: 978-0136905691

13. Ivor Horton and Peter Van Weert ; Beginning C++20: From Novice to Professional; Apress; 6th ed.; 2020; ISBN-13: 978-1484258835
14. Marius Bancila ; Modern C++ Programming Cookbook: Master C++ core language and standard library features, with over 100 recipes, updated to C++20; Packt Publishing ; 2nd Edition; 2020; ISBN-13: 978-1800208988
15. Andreas Fertig, Franziska Panther; Programming with C++20 (grayscale edition): Concepts, Coroutines, Ranges, and more Paperback; Fertig Publications; 2021
16. Rainer Grimm and Beatrix Jaud-Grimm ; C++20: Get the Details; Independently published; 2021; ISBN-13: 979-8732989458
17. Slobodan Dmitrović; Modern C++ for Absolute Beginners: A Friendly Introduction to C++ Programming Language and C++11 to C++20 Standards; Press; 1st ed. edition; 2020; ISBN-13: 978-1484260463
18. Ray Lischner; Exploring C++20: The Programmer's Introduction to C++; Apress; 3rd ed. edition; 2020; ISBN-13: 978-1484259603

-----C++ Avançado

1. Scott Meyers; Effective Modern C++ : 42 Specific Ways to Improve Your Use of C++11(1rd Edition); O'Reilly; 2014;
2. Marc Gregoire; Professional C++; 5nd; Wrox; 2021;
3. Boris Schaling; The Boost C++ Libraries; 2nd; 2014
4. Herb Sutter; C++ Coding Standards; Pearson; 2005;
5. Scott Meyers; C++ moderno e eficaz: 42 formas específicas de aprimorar seu uso de C++11 e C++14; Alta Books; 1ª edição; 2014;
6. Scott Meyers; Effective C++ : 55 Specific Ways to Improve Your Programs and Designs; 3 Edition; Addison-Wesley; 2005;
7. Scott Meyers; More Effective C++; Addison-Wesley; 2006;
8. Martin Fowler; Refatoração: Aperfeiçoando o Design de Códigos Existentes; Novatec Editora; 2ª edição; 2020;
9. Robert C; Martin; Código limpo: Habilidades práticas do Agile Software; Alta Books; 1ª edição; 2009;
10. Peter Gottschling; Discovering Modern C++: An Intensive Course for Scientists, Engineers, and Programmers (C++ In-Depth); Addison-Wesley Professional; 1ª Edição; 2015;
11. Davide Di Gennaro; Advanced Metaprogramming in Classic C++; Edit; Apress; 3ª edição; 2015;

-----STL e Templates

1. Marius Bancila ; Template Metaprogramming with C++: Unlock the power of template metaprogramming to write robust and efficient programs; Packt Publishing ; 2022; ISBN-13: 978-1803243450
2. Jacek Galowicz; C++17 STL Cookbook; Packt Publishing; 2017;
3. David R; Musser; STL Tutorial and Reference Guide: C++ Programming with the Standard Template Library; Addison-Wesley; 2010;
4. Nicolai M; Josuttis; The C++ Standard Library: A Tutorial and Reference; Addison-Wesley Pub Co; 2 edition; 2012;
5. Pete Becker; C++ Standard Library Extensions; Addison-Wesley; 2007;
6. Bjorn Larlsson; Beyond the C++ standard library; Addison-Wesley; 2006;
7. David Vandevoorde; C++ templates meta programming; Addison-Wesley; 2005
8. Andrei Alexandrescu; Modern C++ Design; Addison-Wesley; 2001;
9. Angelika Langer; Standard C++ iostreams and locales; Addison-Wesley; 2000;

-----Multiplataforma (GNU/Linux)

1. Richard E. Silverman; Git: Guia Prático ; Novatec Editora; 1ª edição; 2019;
2. Mariot Tsitoara; Beginning Git and Github: A Comprehensive Guide to Version Control, Project Management, and Teamwork for the New Developer; Apress; 2019;
3. Kenneth Geisshirt, Emanuele Zattin, Aske Olsson; Git Version Control Cookbook: Leverage version control to transform your development workflow and boost productivity; Packt Publishing; 2 Edition; 2018;
4. Radovan Bast, Roberto Di Remigio; CMake Cookbook: Building, testing, and packaging modular software with modern Cmake; Packt Publishing; 1ª edição; 2018;
5. Rafal Swidzinski ; Modern CMake for C++: Discover a better approach to building, testing and packaging your software; Packt Publishing; 1ª Edition; 2022;

-----Bibliotecas e interfaces gráficas

1. Nibedit Dey; Cross-Platform Development with Qt 6 and Modern C++: Design and build applications with

modern graphical user interfaces without worrying about platform dependency; Packt Publishing; 1ª edição; 2021;

2. Mark Summerfield; Advanced Qt Programming: Creating Great Software with C++ and Qt 4; Addison-Wesley Professional, 2010;
3. Jasmim Blanchete; C++ GUI Programming With Qt 4; Prentice Hall; 2006;
4. Alan Ezust; An introduction to design patterns in C++ with Qt 4; Prentice Hall; 2006;
5. Jasmin Blanchette and Mark Summerfield; “C++ GUI Programming With QT 3”; Prentice Hall; 2004;
6. Marcelo Cohen; Open GL uma abordagem prática; Novatec; 2006;

-----Otimização e processamento paralelo e distribuído

1. Anthony Williams; C++ Concurrency in Action: Practical Multithreading (Paperback) ; Manning Publications; 2nd; 2017;
2. Maya Posch ; Mastering C++ Multithreading; Packt Publishing; 2017;
3. Björn Andrist, Viktor Sehr; C++ High Performance: Master the art of optimizing the functioning of your C++ code, 2nd Edition; 2020;
4. Cameron Hughes e Tracey Hughes; Professional Multicore Programming; Wrox, 2008;
5. Cameron Hughes e Tracey Hughes; Parallel and Distributed Programming Using C++; Addison-Wesley, 2004;
6. H.S. Stone; High performance computer architecture; Addison-Wesley; 1992;
7. Foster; Designing and building parallel programs; Massachusetts; Addison Wesley; 3ª edição; 1993;
8. Peter Pacheco; Parallel Programming With MPI; Morgan Kaufmann Publishers; 1996;
9. David R; Butenhof; Programming with POSIX(R) Threads; Addison-Wesley, 1987;
10. George Em Karniadakis , Robert M; Kirby II ; Parallel Scientific Computing in C++ and MPI: A Seamless Approach to Parallel Algorithms and their Implementation; Cambridge University Press; 2003;
11. Yair Shapira; Solving PDEs in C++; SIAM, Society for Industrial and Applied Mathematics; 2006;
12. B; H; Flowers; An Introduction to Numerical Methods in C++; Oxford University Press; USA; Rev Sub edition; 2000;
13. Jack Dongarra et al; Numerical Linear Algebra for High-Performance Computers; SIAM; 1998;
14. Paide Keyes et al; Domain Based Parallelism and Problem Decomposition Methods in Computational Science and Engineering; SIAM; 2009;
15. Marcos Pitanga; Computação em Clusters; Brasport; 2003;
16. Michel A; Henoux et. al; Parallel Processing for Scientific Computing; SIAM; 2006;
17. Daniel J; Duffy; Monte Carlo Frameworks: Building Customisable High-performance C++ Application; Wiley; 2009;

-----Aplicações científicas e engenharia

1. James Paul Holloway; Introdução a Programação para Engenharia; LTC, 2006;
2. John J; Barton, Lee R; Nackman; Scientific and Engineering C++: An Introduction With Advanced Techniques and Examples; Addison-Wesley Pub Co; 1st edition; 1994;
3. Daoqi Yang ; C++ and Object-oriented Numeric Computing for Scientists and Engineers Springer Verlag; 1st edition; 2000;
4. Yair Shapira; Solving PDEs in C++: Numerical Methods in a Unified Object Oriented Approach; SIAM; 2006;
5. Gilbert Strang; Computational Science and Engineering; SIAM; 2007;
6. Jorge J. Moré and Stephen J. Wright; Optimization Software Guide; SIAM; 1993;
7. Yorick Hardy; Computer Algebra With Symbolic C++ (Paperback (Editor) World Scientific Publishing Company; 2008;
8. Marcia Ruggiero; Cálculo numérico; Pearson, 1996, 2.ed; 8534602042
9. Chen, Zhang Xin; Huan, Guaren; Computational Methods for Multiphase Flow in Porous Media; Edt: SIAM;
10. Walter Savitch; Problem solving with C++; pearson; 2007;
11. Zhangxin Chen; Computational Methods for Multiphase Flows in Porous Media ; 2006;
12. D.M.Capper; C++ for scientists, engineers and mathematicians; Springer; 2001;
13. Daniel J; Duffy, Joerg Kienitz; Monte Carlo Frameworks: Building Customisable High-performance C++ Applications; Wiley; Har/Cdr edition; 2009;
14. Allen B; Downey; How To Think Like A Computer Scientist: C++ Version; CreateSpace; 2009;
15. Mark Lee; C++ Programming for the Absolute Beginner; Course Technology PTR; 2.ed; 2009;
16. Robert Sedgewick; Algorithms in C++; Addison-Wesley Professional; 1.ed; 2009;
17. Gary J; Bronson; C++ for Engineers and Scientists; Course Technology; 3 edition; 2009;
18. Damon Danieli; C++ Algorithms for Digital Signal Processing, 2/e, Format: Kindle Edition ; Publisher:

Prentice Hall Professional; 2.ed; 2008;

19. Edward Scheinerman; C++ for mathematicians; CRC; 2006;
20. Armando de Oliveira Fortuna; Técnicas Computacionais para Dinâmica dos Fluidos: Conceitos Básicos e Aplicações; Editora da Universidade de São Paulo – EDUSP; São Paulo SP; Primeira, 85-314-0526-2; 2000;
21. Randall L; Eubank; Statistical Computing in C++ and R; Chapman & Hall/CRC; 2010;
22. Neil Smyth. Fedora 31 Essentials: Learn to install, administer and deploy Fedora 31 systems (English Edition); Payload Media, Inc; 2020;

O livro texto a ser utilizado na disciplina:

1. **Bueno; A;D; Programação Orientada a Objeto em C++ - Aprenda a programar em ambiente multiplataforma com software livre; Terceira Edição; Editora Novatec; São Paulo (2025); (previsão);**
2. Bueno. A.D. Apostila “Projeto de Engenharia - Sistema Equipamentos Conectados (interruptor/lâmpada)”, 2024.
3. Bueno. A.D. Apostila “Projeto de Integração Numérica - Integral Trapézio e Simpson”, 2024.
4. Bueno. A.D. Apostila “Projeto de Interpolação”, 2024.
5. Bueno. A.D. Apostila “Projeto de Aquisição de Dados Experimentais”, 2024.
6. Bueno. A.D. Apostila “Projeto de Processamento de Imagens Rochas Digitais”, 2024.