

Especialista em Programação em C++

Programa de Estudo

Duração: 6 meses

Unidade 10: Introdução ao C++ e ambiente de desenvolvimento

História do C++ e diferenças com C e Java
Instalação de compiladores (GCC, Clang, Visual Studio)
Estrutura de um programa C++: includes, namespace, main
Compilação e execução a partir do IDE e linha de comando

Unidade 20: Tipos de dados, variáveis e operadores

Tipos primitivos: int, float, double, char, bool
Variáveis locais, globais e constantes (const, constexpr)
Operadores aritméticos, relacionais e lógicos
Conversões de tipo implícitas e explícitas (casting)

Unidade 30: Controle de fluxo e funções

Condicionais: if, else if, switch
Laços: for, while, do-while, break, continue
Declaração e definição de funções, passagem por valor e referência
Recursividade: fatorial, Fibonacci e algoritmos recursivos

Unidade 40: Arrays, ponteiros e gestão de memória

Arrays estáticos e dinâmicos, matrizes bidimensionais
Ponteiros: declaração, aritmética e ponteiros para funções
Gerenciamento de memória dinâmica: new, delete e smart pointers
Referências vs ponteiros: quando usar cada um

Unidade 50: Programação Orientada a Objetos em C++

Classes e objetos: construtores, destrutores e encapsulamento
Herança simples e múltipla, polimorfismo com funções virtuais
Sobrecarga de operadores e funções friend
Classes abstratas e interfaces em C++

Unidade 60: Templates e Standard Template Library (STL)

Funções e classes template, especialização de templates
Containers STL: vector, list, map, set, unordered_map
Iteradores e algoritmos STL (sort, find, transform)
Expressões lambda e programação funcional em C++11/17

Unidade 70: Manipulação de arquivos e exceções

Fluxos de E/S: cin, cout, fstream para arquivos de texto e binários
Tratamento de exceções: try, catch, throw e exceções personalizadas
RAII (Resource Acquisition Is Initialization)
Boas práticas de tratamento de erros em C++

Unidade 80: Projeto integrador: aplicação C++ de alta performance

Design de arquitetura para sistemas C++ complexos
Otimização de desempenho: profiling e análise de gargalos
Padrões de design: Singleton, Factory, Observer em C++
Desenvolvimento e apresentação do projeto final de certificação

