

Aula 01: Introdução à Programação, Lógica e IDE (OnlineGDB)

Programação: é o processo de escrever instruções que o computador pode executar. Com ela, criamos sites, apps, jogos, sistemas operacionais e mais.

Lógica de Programação: é a base do raciocínio para resolver problemas com passos bem definidos. Ex: fazer um café envolve sequência lógica (esquentar água, colocar pó, coar, etc).

Linguagem C: é uma linguagem de propósito geral, criada nos anos 70, que influencia muitas linguagens modernas (como C++, Java, etc).

IDE OnlineGDB: é um ambiente online gratuito para codar, compilar e executar códigos em C sem precisar instalar nada. Ideal para iniciantes.



Aula 02: Primeiro (“Hello World”);

- `#include <stdio.h>`: importa biblioteca padrão de entrada/saída.
- `int main()`: ponto inicial do programa.
- `printf`: imprime na tela.
- `return 0`: encerra o programa corretamente.
- Execução: no OnlineGDB, clicar em “Run” compila e executa o programa.



Aula 03: Variáveis

- `#include <stdio.h>`: importa biblioteca padrão de entrada/saída.
- `int main()`: ponto inicial do programa.
- `printf`: imprime na tela.
- `return 0`: encerra o programa corretamente.
- Execução: no OnlineGDB, clicar em “Run” compila e executa o programa.



Aula 04: Operadores lógicos

Ensinar como fazer operações matemáticas, comparações e tomadas de decisão.

- **Aritméticos:** +, -, *, /, %
 - Ex: `int soma = 2 + 3;`
- **Relacionais:** ==, !=, <, >, <=, >=
 - Usados em condições. Ex: `if (x > y)`
- **Lógicos:** && (E), || (OU), ! (NÃO)
 - Ex: `if (x > 0 && y < 10)`
- **Atribuição composta:** +=, -=, *=, /=, %=
 - Ex: `x += 2;` equivale a `x = x + 2;`



Aula 05: Entrada e Saída de Dados `printf`, `scanf`

- Saída de dados com `printf`:
 - Ex: `printf("Idade: %d", idade);`
- Entrada com `scanf`:
 - Ex: `scanf("%d", &idade);`
- O `&` indica o endereço de memória da variável.
- Especificadores:
 - `%d` → inteiro
 - `%f` → número com ponto
 - `%c` → caractere
 - `%s` → string



Aula 06: Estruturas Condicionais

if, else, switch-case

- A estrutura condicional if permite executar um bloco de código somente se uma condição for verdadeira.
- A cláusula else serve para executar outro bloco quando a condição do if for falsa.
- O else if permite verificar múltiplas condições diferentes, em sequência, facilitando decisões mais complexas.
- Exemplos práticos mostram como usar essas estruturas para, por exemplo, verificar se um número é positivo, negativo ou zero, ou determinar se um número é par ou ímpar.
- Essas estruturas são essenciais para controlar o fluxo do programa, tornando-o dinâmico e responsivo a diferentes entradas.



Aula 07: Estruturas Condicionais

Do, While, For

- O laço while executa o código enquanto a condição é verdadeira. Se a condição for falsa na primeira verificação, o código não será executado nem uma vez.
- O do...while executa o código pelo menos uma vez antes de testar a condição, garantindo que o bloco rode uma vez sempre.
- O for é usado para loops com contador conhecido, executando o código por um número pré-definido de vezes.
- Exemplos incluem contadores simples que imprimem valores de 0 a 4 usando cada um dos laços, para mostrar a diferença entre eles.
- Entender essas estruturas é fundamental para criar programas que realizam tarefas repetitivas de forma eficiente e controlada.



Aula 08: Funções

- **Definição:** Funções são blocos de código nomeados que executam uma tarefa específica e podem ser chamados várias vezes dentro do programa.
- **Declaração e Definição:** Toda função deve ser declarada (protótipo) e definida com seu corpo de código.
- **Parâmetros:** Funções podem receber parâmetros para trabalhar com dados fornecidos externamente.
- **Valor de Retorno:** A função pode retornar um valor ao código que a chamou, usando o comando return.
- **Funções void:** Funções que não retornam valor, usadas para executar ações sem entregar resultado direto.
- **Vantagens:**
 - Evita repetição de código.
 - Facilita a manutenção e a organização do programa.
 - Permite dividir problemas complexos em partes menores e mais simples.

