



# Introdução a Linguagem de Programação C++

---

Prof. Tiago Gonçalves Botelho



# Breve histórico

---

- Em 1973, Dennis Ritchie reescreveu o sistema operacional UNIX em uma linguagem de alto nível.
- Por volta de 1980, existiam diversas versões de compiladores C oferecidas por várias empresas, compatíveis com outros Sistemas Operacionais.
- No início dos anos 80 Bjarne Stroustrup desenvolveu a linguagem C++. Trata-se de uma extensão da linguagem C, com várias características aperfeiçoadas.



# C++ comparado a outras linguagens

---

- C++ é uma linguagem de alto nível, e assembly uma linguagem de baixo nível por estar próxima da linguagem da máquina.
- Infelizmente, quanto mais clara uma linguagem for para o humano (simplicidade >), mais obscura será para a máquina (velocidade <).



# Aplicações escritas em C/C++

---

- A maioria dos Sistemas Operacionais;
- Planilha excel;
- Banco de dados: dBase III, IV e Access;
- Aplicações gráficas: Efeitos especiais de filmes como Star Trek e Star War.

# Características da linguagem C/C++



---

- Portabilidade entre máquinas e Sistemas Operacionais;
- Alia características de linguagens de alto e baixo nível;
- Programas estruturados ou orientados à objeto;
- Total interação com o sistema operacional;
- Código compacto e rápido, quando comparado ao código de outras linguagens de complexidade análoga.
- Linguagem compilada: lê todo o código fonte e gera o código objeto (ling. de máquina) uma única vez.



# Detalhes importantes

---

- Sempre que o código fonte for alterado ele deve ser novamente compilado;
- C++ é “case sensitive”;
- Deve-se listar antecipadamente todas as variáveis utilizadas no programa.



# Palavras reservadas

---

- São palavras que têm um significado especial para a linguagem;

*auto, break, case, if, for, while, do, main, continue, return, const,...*

- C++ entende tais palavras apenas em letras minúsculas.



# Bibliotecas

---

- Conjunto de funções para realizar tarefas específicas;
- Biblioteca padrão C++ – funções básicas;
- As primeiras linhas do programa indicam as bibliotecas utilizadas.

```
#include <nome_biblioteca>
```





# IDE para desenvolvimento de programas

---

- Uma IDE (Integrated Development Environment) é o ambiente de desenvolvimento dos programas.
  - **Code::Blocks C/C++:** Uma das mais usadas IDEs, o Code Blocks é sem dúvida uma boa IDE para se trabalhar com C++. Essa IDE possui o Código Fonte aberto, pode ser baixado no site oficial.
  - **Dev C/C++:** Uma ótima para se trabalhar com C ++.
- Os códigos fonte são salvos com a extensão "nome\_codigo.**cpp**"



# Estrutura de um programa C++

---

```
1  #include<bits/stdc++.h>
2
3  int main() {
4      printf("Bem vindo a C++!\n");
5      return 0;
6  }
```



# Estrutura de um programa C++

---

```
1  /* Um primeiro programa em C++
2     Bem vindo ao C++
3     Comentário de mais de uma linha*/
4  #include<bits/stdc++.h> //Biblioteca padrão para diversos comandos
5
6  int main() // o main é uma função obrigatória para todo programa em C++ e o int significa que a função devolve um inteiro
7  { // indica começar o corpo de uma função, termina com o } observado na linha 10
8     printf("Bem vindo ao C++!\n"); //instrui o computador a imprimir na tela a mensagem a seguir entre aspas
9     return 0; // é incluída no fim de toda função main, o valor zero indica que o programa terminou com sucesso
10 }
```



# Variáveis em C++

Quanto a declaração dos tipos de dados:

Pseudocódigo	Linguagem C++
inteiro	int
real	float, double
literal	char ou char [n]
logico	bool



# Exemplos de declaração

---

```
double x;
```

```
char sexo, nome[30];
```

```
int i;
```

```
bool j;
```



# Estrutura sequencial

---

- Pseudocódigo:

```
ALGORITMO somatorio;  
DECLARE N1, N2, N3, SOMA: INTEIRO  
INICIO  
    ESCREVA ("Digite o primeiro valor: ")  
    LEIA(N1)  
    ESCREVA ("Digite o segundo valor: ")  
    LEIA(N2)  
    ESCREVA ("Digite o terceiro valor: ")  
    LEIA(N3)  
    SOMA ← N1 + N2 + N3  
    ESCREVA ("A soma é: ", SOMA)  
FIM.
```



# Estrutura sequencial

## ■ Código em C++ baseado no pseudocódigo

```
1  #include<bits/stdc++.h>
2
3  int main() {
4      int n1,n2,n3,soma;
5      printf("Digite o primeiro valor: ");
6      scanf("%d",&n1);
7      printf("Digite o segundo valor: ");
8      scanf("%d",&n2);
9      printf("Digite o terceiro valor: ");
10     scanf("%d",&n3);
11     soma=n1+n2+n3;
12     printf("A soma e: %d\n",soma);
13     return 0;
14 }
```



# Estrutura sequencial

Código em C++ baseado no pseudocódigo sem comandos de saída junto à entrada de dados

```
1      #include<bits/stdc++.h>
2
3      int main() {
4          int n1,n2,n3,soma;
5          scanf("%d",&n1);
6          scanf("%d",&n2);
7          scanf("%d",&n3);
8          soma=n1+n2+n3;
9          printf("A soma e: %d\n",soma);
10         return 0;
11     }
```

Obs.: Este é o modelo de entrada de dados dos exercícios online.





# Estrutura sequencial

---

- Para leitura de dados, ao usar o comando scanf utiliza-se para cada um dos tipos de dados:
- Na escrita dos resultados na tela, utiliza-se os mesmos códigos.

Código	Significado
%d	int
%lf	double
%c	char
%s	char [n]
%%	Coloca na tela um %



# Aritmética: Operadores

Operação em C++	Operador Aritmético	Expressão Algébrica	Expressão C++
Adição	+	$f + 7$	$f + 7$
Subtração	-	$p - c$	$p - c$
Multiplicação	*	$bm$	$b * m$
Divisão	/	$x/y$ ou $x:y$	$x / y$
Módulo	%	$r \bmod s$	$r \% s$



# Aritmética: Precedência dos operadores aritméticos

Operador(es)	Operação(ões)	Ordem de avaliação (precedência)
( )	Parênteses	Calculados primeiro. Se os parênteses estão aninhados, primeiro é calculado o par mais interno na expressão. Se houver vários pares de parênteses “no mesmo nível” (isto é, não aninhados), eles são calculados da esquerda para a direita.
*, / ou %	Multiplicação Divisão Módulo	Calculados em segundo lugar. Se houver vários, eles são calculados da esquerda para a direita.
+ ou -	Adição Subtração	Calculados por último. Se houver vários, eles são calculados da esquerda para a direita.



# Tomada de decisões: os operadores relacionais e de igualdade

**Operadores de igualdade algébricos padrão e operadores relacionais**

**Operador C++ de igualdade ou relacional**

**Exemplo de condição em C++**

**Significado da condição em C++**

## **Operadores de igualdade:**

>	>	$x > y$	<b>x</b> é maior que <b>y</b>
<	<	$x < y$	<b>x</b> é menor que <b>y</b>
$\geq$	$\geq$	$x \geq y$	<b>x</b> é maior ou igual a <b>y</b>
$\leq$	$\leq$	$x \leq y$	<b>x</b> é menor ou igual a <b>y</b>

## **Operadores relacionais:**

=	==	$x == y$	<b>x</b> é igual a <b>y</b>
$\neq$	!=	$x != y$	<b>x</b> não é igual a <b>y</b>



# Bibliografia

---

- Deitel, H.M.; Deitel P.J.; C++: como programar. Editora Bookman, 2001.