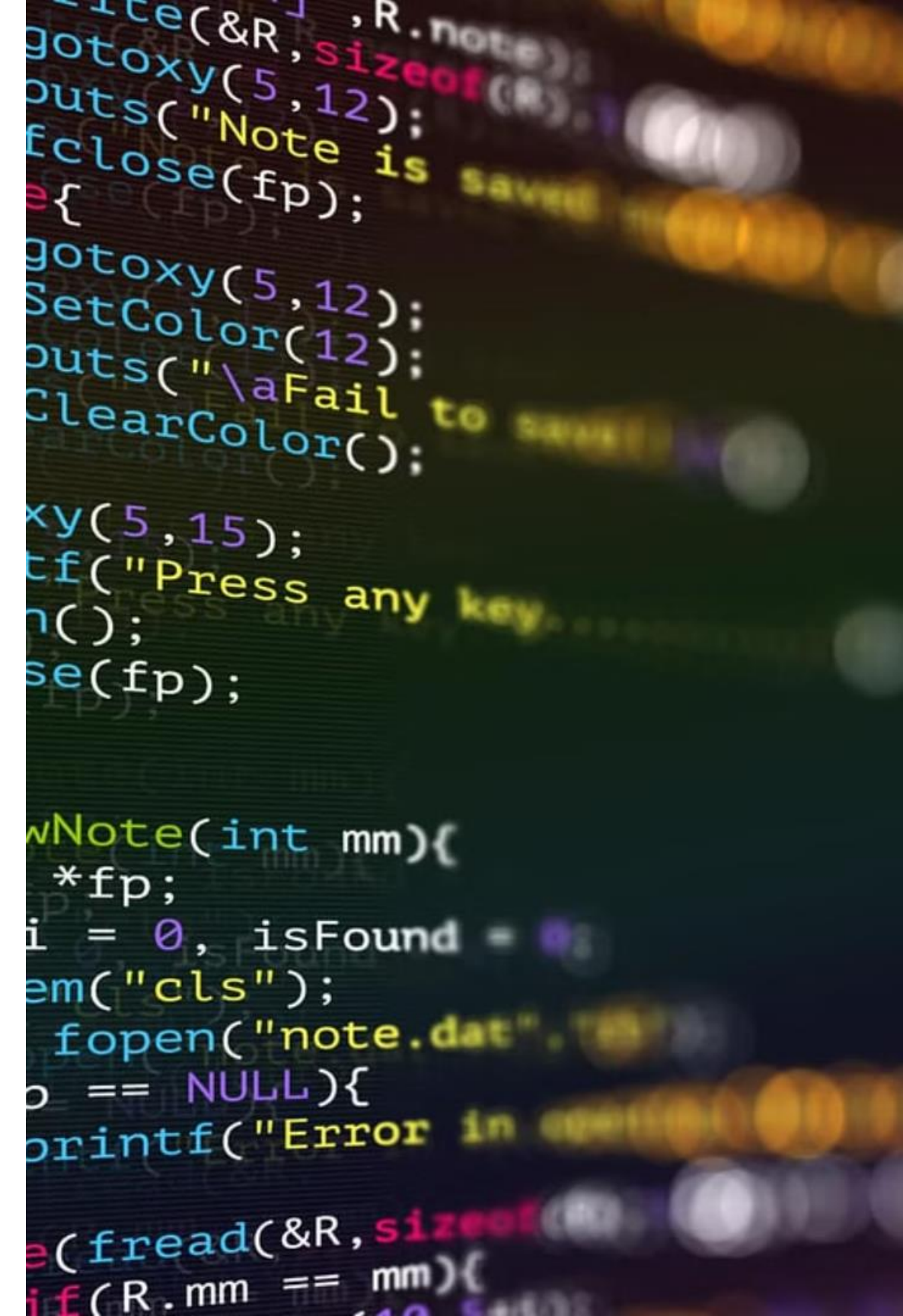


Introdução a Linguagem C++

Disciplina: Programação Aplicada

Prof. Rodrigo C. N. Maciel



Conteúdo

- Introdução ao C++
- Entrada/Saída de Terminal
- Variáveis, Expressões e Declarações de Atribuição
- Tipos de Dados
- Operadores;
- Exercícios de Fixação

Introdução ao C++

C++ é uma linguagem de programação de propósito geral que é amplamente utilizada para desenvolver sistemas de software, aplicativos, jogos e muito mais;

Foi desenvolvida no Bell Laboratories no início da década de 1980 por **Bjarne Stroustrup** como uma extensão da linguagem C;

Incorpora características de programação orientada a objetos, além de recursos de programação genérica e programação de baixo nível.



Entrada/Saída de Terminal

A entrada e saída de dados em C++ são frequentemente realizadas usando a biblioteca padrão '**iostream**', que fornece os objetos '**cin**' para entrada e '**cout**' para saída.

```
#include <iostream>

int main() {
    int num;
    std::cout << "Digite um número: "; // Saída
    std::cin >> num; // Entrada
    std::cout << "Você digitou: " << num << std::endl; // Saída
    return 0;
}
```

Neste exemplo:

- '**#include <iostream>**' inclui a biblioteca de entrada/saída padrão.
- '**std::cout**' é usado para imprimir texto no terminal.
- '**std::cin**' é usado para ler a entrada do usuário.
- '**std::endl**' adiciona uma nova linha na saída.

Entrada/Saída de Terminal

Uma alternativa menos verbosa é utilizando o comando '**using namespace std**'. Ele é usado para evitar a necessidade de prefixar nomes de elementos da biblioteca padrão com '**std: :**'. Em C++, a biblioteca padrão (Standard Library) está dentro do namespace '**std**'. Isso inclui componentes como '**std: :cout**', '**std: :cin**', '**std: :string**', entre outros.

```
#include <iostream>

using namespace std;

int main() {
    int num;
    cout << "Digite um número: "; // Saída
    cin >> num; // Entrada
    cout << "Você digitou: " << num << endl; // Saída
    return 0;
}
```


Variáveis, Expressões e Declarações de Atribuição

Em C++, variáveis são usadas para armazenar dados.

A declaração de variáveis especifica seu tipo e nome.

A atribuição é usada para definir o valor de uma variável.

```
#include <iostream>

int main() {
    int a = 5; // Declaração e inicialização de uma variável inteira
    double b = 3.14; // Declaração e inicialização de uma variável de ponto flutuante
    char c = 'A'; // Declaração e inicialização de uma variável de caractere

    int soma = a + 2; // Expressão e atribuição
    double area = b * a; // Expressão e atribuição

    std::cout << "Soma: " << soma << std::endl;
    std::cout << "Área: " << area << std::endl;
    std::cout << "Caractere: " << c << std::endl;

    return 0;
}
```

Neste exemplo:

- **int**, **double** e **char** são tipos de dados.
- **a**, **b** e **c** são variáveis.
- As expressões **a + 2** e **b * a** são usadas para calcular valores e atribuí-los a **soma** e **area**, respectivamente.

Tipos Numéricos

Use **int** para armazenar números inteiros, e **float** ou **double** para números com casas decimais.

Precisão de Ponto Flutuante

float tem precisão de 6-7 dígitos, enquanto **double** tem cerca de 15 dígitos, sendo mais seguro para cálculos.

Exemplos

```
int myNum = 1000; float myNum = 5.75; double myNum = 19.99;
```

Tipos de dados básicos do C++

Tipos Booleanos

Um tipo de dados booleano é declarado com a palavra-chave **bool** e só pode assumir os valores **verdadeiro** ou **falso**.

Quando o valor é retornado, **verdadeiro = 1** e **falso = 0**.

Exemplos:

```
bool isCodingFun = true;
```

```
bool isFishTasty = false;
```

```
cout << isCodingFun; // Outputs 1 (true)
```

```
cout << isFishTasty; // Outputs 0 (false)
```


Tipos de dados básicos do C++

Tipo Caracteres

O tipo de dados char é usado para armazenar um único caractere. O caractere deve estar entre aspas simples, como 'A' ou 'c':

Exemplo:

```
char myGrade = 'B';
```

Tipos de dados básicos do C++

Tipo String

O tipo **string** é usado para armazenar uma sequência de caracteres (texto). Os valores da string devem estar entre aspas duplas.

Exemplo:

```
string greeting = "Hello";
```

Obs: para o seu uso é necessário a importação da biblioteca: **#include** <string>

Mais detalhes sobre tipos de dados em: <https://www.w3schools.com/CPP/>

Operadores Aritméticos

Operator	Name	Description	Example
+	Addition	Adds together two values	$x + y$
-	Subtraction	Subtracts one value from another	$x - y$
*	Multiplication	Multiplies two values	$x * y$
/	Division	Divides one value by another	x / y
%	Modulus	Returns the division remainder	$x \% y$
++	Increment	Increases the value of a variable by 1	$++x$
--	Decrement	Decreases the value of a variable by 1	$--x$

Fonte: https://www.w3schools.com/CPP/cpp_operators.asp

Alguns operadores de atribuição

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3

Lista completa em: https://www.w3schools.com/Cpp/cpp_operators_assignment.asp

Operadores de comparação

Operator	Name	Example
<code>==</code>	Equal to	<code>x == y</code>
<code>!=</code>	Not equal	<code>x != y</code>
<code>></code>	Greater than	<code>x > y</code>
<code><</code>	Less than	<code>x < y</code>
<code>>=</code>	Greater than or equal to	<code>x >= y</code>
<code><=</code>	Less than or equal to	<code>x <= y</code>

Operadores lógicos

Operator	Name	Description	Example
&&	Logical and	Returns true if both statements are true	<code>x < 5 && x < 10</code>
	Logical or	Returns true if one of the statements is true	<code>x < 5 x < 4</code>
!	Logical not	Reverse the result, returns false if the result is true	<code>!(x < 5 && x < 10)</code>

Fonte: https://www.w3schools.com/CPP/cpp_operators_logical.asp

Exercício 1

Crie um programa que peça ao usuário para digitar seu nome e idade. O programa deve então exibir uma mensagem dizendo "Olá, [nome]! Você tem [idade] anos."

Solução:

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    string nome;
    int idade;

    cout << "Digite seu nome: ";
    cin >> nome;
    cout << "Digite sua idade: ";
    cin >> idade;

    cout << "Olá, " << nome << "! Você tem " << idade << " anos." << endl;

    return 0;
}
```

Exercício 2

Crie um programa que solicite ao usuário dois números inteiros. O programa deve então calcular e exibir a soma, subtração, multiplicação e divisão desses números.

Solução:

```
#include <iostream>

using namespace std;

int main() {
    int num1, num2;

    cout << "Digite o primeiro número: ";
    cin >> num1;
    cout << "Digite o segundo número: ";
    cin >> num2;

    cout << "Soma: " << num1 + num2 << endl;
    cout << "Subtração: " << num1 - num2 << endl;
    cout << "Multiplicação: " << num1 * num2 << endl;
    cout << "Divisão: " << num1 / num2 << endl;

    return 0;
}
```

Exercício 3

Crie um programa que leia uma temperatura em graus Celsius do usuário e a converta para Fahrenheit. A fórmula de conversão é $F = C * 9/5 + 32$. Exiba a temperatura convertida.

Solução:

```
#include <iostream>

using namespace std;

int main() {
    double celsius;
    double fahrenheit;

    cout << "Digite a temperatura em graus Celsius: ";
    cin >> celsius;

    fahrenheit = celsius * 9 / 5 + 32;

    cout << "Temperatura em Fahrenheit: " << fahrenheit << endl;

    return 0;
}
```