

**UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
FACULDADE DE COMPUTAÇÃO**



Algoritmos

**Prof. Denis Rosário
Preparado por Pedro Cumino
Email: denis@ufpa.br**



Agenda

Introdução ao C++

- História
- Especificações
- *Hello World!*
- Palavras reservadas
- Variáveis e tipos de dados
- Criação e inicialização de variáveis
- Saída/Entrada de dados
- Operações aritméticas
- Operações relacionais
- Estrutura de decisão
- Estruturas de repetição



História

- C++ foi desenvolvido por Bjarne Stroustrup a partir de 1979 no Bell Labs em Murray Hill, Nova Jersey, como um aprimoramento da linguagem C e originalmente chamado C com Classes, mas depois foi renomeado C++ em 1983



Especificações

- C++ é executado em uma variedade de plataformas, como Windows, Mac OS e as várias versões do UNIX
- C++ é um superconjunto de C, e praticamente qualquer programa válido em C é um programa válido em C++



Especificações

- Compilado (<https://godbolt.org/>)
- Sensível a maiúsculas e minúsculas
- Oferece suporte a programação processual, orientada a objetos e genérica
- É considerada como uma linguagem de nível médio, pois compreende uma combinação de recursos de linguagem de alto nível (longe do código de máquina) e baixo nível (próximo do código de máquina)



Hello world!

```
// meu primeiro programa em C++
/* Um Primeiro Programa */
1.#include <iostream> // biblioteca de entrada e saída
2.using namespace std;
3.// main(): método onde o programa inicia sua execução.
4.int main() {
5.    // imprime Hello World!
6.    cout << "Hello World!" << endl;
7.    return 0;
8.}
```

```
$ g++ hello.cpp
$ ./a.out
Hello World!
```

Primeiros Passos

```
// meu primeiro programa em C++
```

```
#include <iostream>
```

```
using namespace std;
```

```
/* Um Primeiro Programa */
```

```
int main () {
```

```
    cout << "Hello World!" <<  
endl;
```

```
    return(0); // programa terminou  
com sucesso
```

```
}
```

- `#include <iostream.h>`

- ☐ diz ao compilador que ele deve incluir o arquivo-cabeçalho `iostream.h` na hora da linkedição.

- `// meu primeiro programa em C++`

- `/* Um Primeiro Programa */` - Comentários

Primeiros Passos

```
// meu primeiro programa em C++
#include <iostream>
using namespace std;
/* Um Primeiro Programa */
int main () {
    cout << "Hello World!" <<
endl;
    return(0); // programa terminou
com sucesso
}
```

■ `int main()`

- ☐ indica que estamos definindo uma função de nome **main**.
- ☐ Todos os programas em C têm que ter uma função **main**, pois é esta função que será chamada quando o programa for executado.

■ `cout << "Hello World!" << endl;`

- ☐ Instrui o computador para imprimir a string "Bem vindo ao C" na tela

■ `return(0)`

- ☐ Indica ao carregador do programa(shell) que o mesmo rodou sem problemas.



Entrada e Saída de Dados

- Em português:

- `Escreva("A media do aluno=", media);`

- Em C++:

- `cout << "A media do aluno =" << media;`

- `cout` -> saída padrão em vídeo



Entrada e Saída de Dados

- Em portugol

- `leia(n1);`

- Em C++:

- `cin >> n1;`

- `cin` -> entrada padrão - teclado



Palavras reservadas

and	and_eq	asm	auto	bitand	bitor	bool
break	case	catch	char	class	const	const_cast
continue	default	delete	do	double	dynamic_cast	else
enum	explicit	export	extern	false	float	for
friend	goto	if	inline	int	long	mutable
namespace	new	not	not_eq	operator	or	or_eq
private	protected	public	register	reinterpret_cast	return	short
signed	sizeof	static	static_cast	struct	switch	template
this	throw	true	try	typedef	typeid	typename
union	unsigned	using	virtual	void	volatile	wchar_t
while	xor	xor_eq	include			



Variáveis e tipos de dados

Tipo	Nome	Modificadores
Lógico	bool	-
Inteiro	int	unsigned, short, long
Real	float, double	long
Caracter	char	-
Vazio	void	-



Variáveis e tipos de dados

Declaração	Tamanho	Faixa
char	1byte	-128 to 127
unsigned char	1byte	0 to 255
int	4bytes	-2147483648 to 2147483647
unsigned int	4bytes	0 to 4294967295
short int	2bytes	-32768 to 32767
unsigned short int	2bytes	0 to 65,535
long int	8bytes	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
signed long int	8bytes	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
unsigned long int	8bytes	0 to 18,446,744,073,709,551,615
float	4bytes	+/- 3.4e +/- 38 (~7 digits)
double	8bytes	+/- 1.7e +/- 308 (~15 digits)
long double	8bytes	+/- 1.7e +/- 308 (~15 digits)
wchar_t	2 or 4 bytes	1 wide character

Tamanho dos tipos de dados

```
#include <iostream>
using namespace std;

int main() {
    cout << "Size of char : " << sizeof(char) << endl;
    cout << "Size of int : " << sizeof(int) << endl;
    cout << "Size of short int : " << sizeof(short int) << endl;
    cout << "Size of long int : " << sizeof(long int) << endl;
    cout << "Size of float : " << sizeof(float) << endl;
    cout << "Size of double : " << sizeof(double) << endl;
    cout << "Size of wchar_t : " << sizeof(wchar_t) << endl;
    return 0;
}
```



Criação de variáveis

1. var
2. n: inteiro
3. x: real
4. b: logico
5. c: caractere
6. y, z, w: real

```
1  #include <iostream>
2  void main(){
3      int n;
4      double x;
5      bool b;
6      char c;
7      float y, z, w;
8  }
```

Inicialização de variáveis

1. inicio

2. n<- 10

3. x<- 10.4

4. b<- falso

5. c<- "a"

6. y<- 1.4

7. z<- 2

8. w<- 0.4

```
1  #include <iostream>
2
3  void main() {
4      int n = 10;
5      double x = 10.4;
6      bool b;
7      char c = "a";
8      float y = 1.4, z = 2;
9      float w = 0.4;
10     b = false;
11 }
```


Saída básica

1. inicio
2. a<- 10
3. escreva("valor = ", a)
4. escreval("valor = ", a)

```
1  #include <iostream>
2
3  int main(){
4      int a = 10;
5      std::cout << "valor = " << a;
6      std::cout << "valor = " << a << std::endl;
7      return 0;
8  }
```

Entrada básica

1. var
2. a: inteiro
3. inicio
4. leia(a)
5. escreval("valor = ",a)

```
1  #include <iostream>
2
3  int main(){
4      int a;
5      std::cin >> a;
6      std::cout << "valor = " << a << std::endl;
7      return 0;
8  }
```

Entrada de string com espaço

1. var
2. name: caractere
3. inicio
4. name<- "Escreva o nome completo:"
5. escreval(name)
6. leia(name)
7. escreval(name)

```
1  #include <iostream>
2  #include <string>
3  // using namespace std;
4
5  int main() {
6      std::string name = "Escreva o nome completo:";
7      std::cout << name << std::endl;
8
9      getline(std::cin, name);
10
11     std::cout << name << std::endl;
12     return 0;
13 }
```

Operações aritméticas básicas

1. var
2. a,b: inteiro
3. inicio
4. a<-16
5. b<-4
6. escreval(a+b)
7. escreval(a-b)
8. escreval(a*b)
9. escreval(a/b)
10. escreval exp(a,2)
11. escreval raizq(a)

```
1  #include <iostream>
2  #include <math.h>
3  using namespace std;
4
5  int main(){
6      int a = 16, b = 4;
7
8      cout << a+b << endl;
9      cout << a-b << endl;
10     cout << a*b << endl;
11     cout << a/b << endl;
12     cout << pow(a,2) << endl;
13     cout << sqrt(a) << endl;
14
15
16     return 0;
17 }
```



Operadores relacionais

Operadores	Sintaxe
Maior que	>
Menor que	<
Maior igual a	>=
Menor igual a	<=
Igual a	==
Diferente	!=

Estrutura de decisão : if/else

■ Em portugol:

```
se (condição) {  
    .... }  
senao{  
    .....}  
-----
```

Exemplo

```
Se(media >=8){  
    Imprima "Aluno Aprovado";  
}  
Senao{  
    Imprima "Aluno reprovado";  
}
```

■ Em C:

```
if (condição){  
    ...}  
else {  
    ...  
}
```

exemplo

```
if (media >=8){  
    std::cout << "aluno aprovado";  
}  
else{  
    std::cout << "aluno reprovado";  
}
```

Estrutura de decisão : if/else

```
1.  var
2.    comprar: logico
3.    preco, dinheiro: real
4.  inicio
5.    dinheiro <- 200
6.    preco <- 150
7.    comprar <- dinheiro >= preco
8.    se(comprar) entao
9.      escreval("Você tem dinheiro suficiente. Compre!")
10.   senao
11.     escreval("Você está duro! Sem chance!")
12.   fimse
```

```
1  #include <iostream>
2  #include <math.h>
3  using namespace std;
4
5  int main(){
6      bool comprar;
7      double preco, dinheiro;
8
9      dinheiro = 200;
10     preco = 150;
11     comprar = (dinheiro >= preco);
12     if(comprar){
13         cout << "Você tem dinheiro
14             suficiente. Compre!" << endl;
15     }
16     else{
17         cout << "Você está
18             duro! Sem chance!" << endl;
19     }
20     return 0;
21 }
```



Estrutura de decisão: if/else

- 1. Escreva um algoritmo que verifica se um número é negativo, positivo ou igual a zero.
- 2. Escreva um algoritmo que encontre o máximo e o mínimo entre dois números reais.
- 3. Escreva um algoritmo que mostre em ordem 3 números inteiros.

Estrutura de decisão : if/else

■ 1.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n = 0;
6
7      cout << "Escreva um número:" << endl;
8      cin >> n;
9
10     if(n < 0){
11         cout << "Número é negativo \n";
12     }
13     else if(n == 0){
14         cout << "Número é zero \n";
15     }
16     else{
17         cout << "Número é positivo \n";
18     }
19     return 0;
20 }
```

Estrutura de decisão : if/else

■ 2.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int a = 0, b = 0;
6
7      cout << "Escreva um valo para 'a':" << endl;
8      cin >> a;
9      cout << "Escreva um valo para 'b':" << endl;
10     cin >> b;
11     if(a > b){
12         cout << a << " É maior que " << b << endl;
13     }
14     else if(a == b){
15         cout << a << " É igual " << b << endl;
16     }
17     else{
18         cout << a << " É menor que " << b << endl;
19     }
20
21     return 0;
22 }
```



Exercícios

1. Escreva um algoritmo que verifica se um número é par ou ímpar.
2. Escreva um algoritmo que verifica se um número é divisível por 5 e 11 ou não.
3. Escreva um algoritmo que verifica se um ano é bissexto ou não.
4. Escreva um algoritmo que encontre todos as raízes de uma equação do segundo grau.
5. Escreva um algoritmo que receba o valor da temperatura e informe se está frio,
agradável ou quente. Frio para temperaturas menores que 18,
Agradável para
temperaturas entre 18 e 28 graus e quente para valores maiores que 28 graus.



Estruturas de repetição: *while*

1. enquanto (*<condição de parada>*) faça
2. *<comandos>*
3. fimenquanto

1. while(*<condição de parada>*){
2. *<comandos>*
3. }



Estruturas de repetição: *while*

1. enquanto (a < 5) faça
2. escreval(a)
3. a<-a+1
4. fimenquanto

1. while(a < 5){
2. cout << a << endl;
3. a++;
4. }

Exercício:

- Escrever programa em C++ que faça mostre na tela os números inteiros de 100 a -100 com intervalo fechado.

Estruturas de repetição: *while*

■ Possível solução:

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int i = 100;
6
7      while(i >= -100){
8          cout << i << endl;
9          i--;
10     }
11 }
```



Estruturas de repetição: *for*

1. para *<variável>* de *<valor-inicial>* ate *<valor-limite>* [passo *<incremento>*] faca
2. *<comandos>*
3. fimpara

1. for(*<variável de inicialização>*; *<condição de parada>*; *<incremento>*){
2. *<comandos>*
3. }



Estruturas de repetição: *for*

1. para *i* de 0 ate 5 passo[1] faça
2. escreval(2*i)
3. fimpara

1. for(int i = 0; i < 5; i++){
2. cout << 2*i << endl;
3. }

Exercício:

- Escrever programa em C++ que mostre os números ímpares de -50 a 50

Estruturas de repetição: *for*

■ Possível solução:

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     for(int i = -49; i < 50; i+=2){
6         cout << i << endl;
7     }
8 }
```



Exercícios

1. Informe se um número digitado é primo. Caso não for, informe por quais números ele é divisível.
2. Digitados dois números (base **b** e expoente **e**), calcule o resultado de b^e utilizando apenas multiplicações.
3. Calcule a soma de todos os números primos existentes entre 1 e 100.
4. Faça o mesmo que na questão **2**, mas usando apenas somas.