

TECNOLOGIAS DE PROGRAMAÇÃO DE SISTEMAS DE
INFORMAÇÃO

Programação C++

PROGRAMAÇÃO ORIENTADA A OBJECTOS | Prof. Doutora Frederica Gonçalves

Cofinanciado por:



UNIDADE CURRICULAR : PROGRAMAÇÃO ORIENTADA A OBJECTOS

- **Origens da linguagem C e C++**

- A linguagem C teve origem no início da década de 70, mais precisamente nos laboratórios Bell da companhia de telecomunicações AT&T.
- A linguagem C++ é um suplemento da linguagem C, que vem acrescentar novos elementos que não existiam nesta mesma linguagem.
- Tem como objectivo:
 - facilitar a complexa tarefa de programação dos sistemas operativos.
 - Exemplo: Sistema operativo Unix.

UNIDADE CURRICULAR : PROGRAMAÇÃO ORIENTADA A OBJECTOS

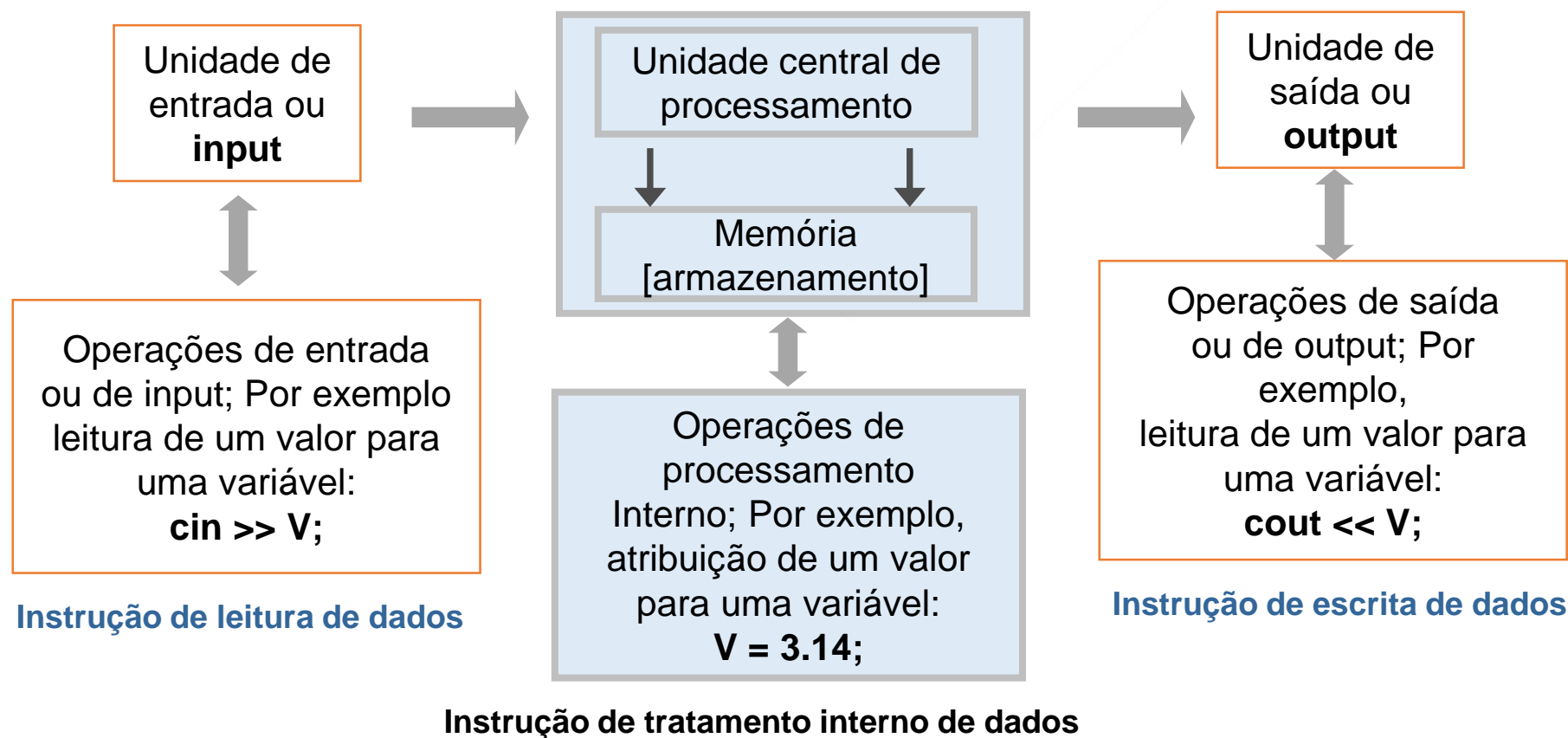
• Exemplo de um programa

- `#include` - indica as bibliotecas que serão utilizadas;
- `main` – indica o início de execução de um programa
- `{ }` – indica respectivamente o início e o fim de um bloco de instruções;
- `printf (" ")` - permite escrever no ecrã o que esta dentro ("");
- `\n`- indica mudança de linha;
- `//` - indica comentários até ao fim dessa linha;
- `/* */`- indica comentários, podendo ocupar várias linhas

```
/*exemplo  
Hello World!*/  
  
#include <stdio.h>  
main () //começo da função  
{  
printf ("\nHello World!\n");  
}
```

UNIDADE CURRICULAR : PROGRAMAÇÃO ORIENTADA A OBJECTOS

• Operações básicas e instruções



UNIDADE CURRICULAR : PROGRAMAÇÃO ORIENTADA A OBJECTOS

- Directivas de compilação

#include <iostream>



Indica ao compilador para incluir uma biblioteca de funções padrão do C++, estando esta ligada às operações de **input** e **output** de fluxo de dados.

using namespace std;



Faz parte de um procedimento usual e é um complemento da directiva anterior tornando visíveis as funções standard.

main ()



Parte do programa, que indica o início da função principal, sendo o ponto onde começa a execução do programa.

UNIDADE CURRICULAR : PROGRAMAÇÃO ORIENTADA A OBJECTOS

- **Declaração de variáveis**

- Declaração da variável **a**, sendo esta do tipo **int**;
- Declara-se primeiro o tipo de dados e só depois a, ou as variáveis;
- Se pretendermos declarar mais de um determinado tipo, basta separá-las por virgulas devendo sempre terminar com ponto e virgula.

Ex.: **int a, b, c;**

```
int a;
```

UNIDADE CURRICULAR : PROGRAMAÇÃO ORIENTADA A OBJECTOS

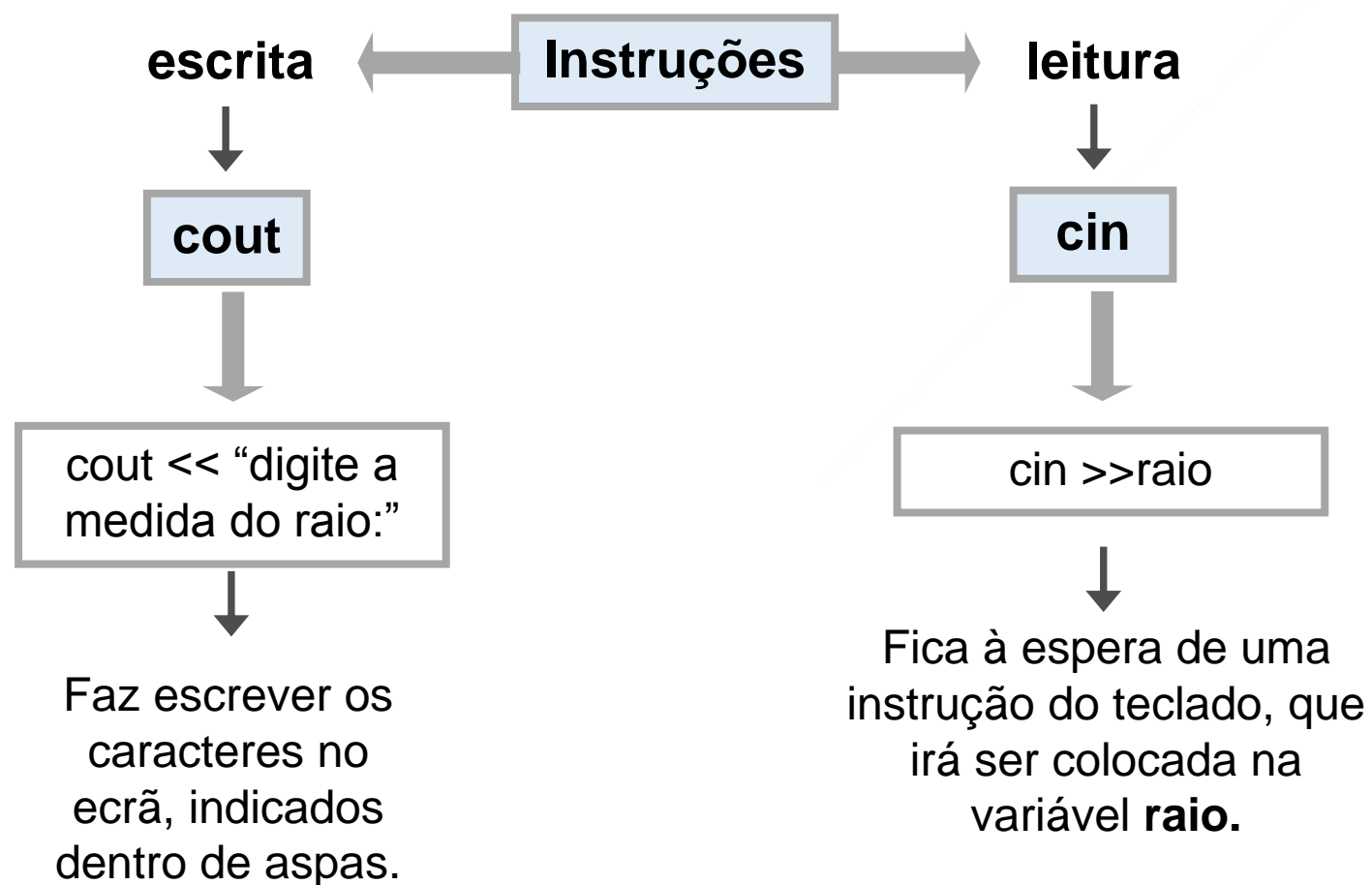
- **Declaração de variáveis**

- Declaração da variável **a**, sendo esta do tipo **int**;
- Declara-se primeiro o tipo de dados e só depois a, ou as variáveis;
- Se pretendermos declarar mais de um determinado tipo, basta separá-las por virgulas devendo sempre terminar com ponto e virgula.

Ex.: **int a, b, c;**

```
int a;
```

UNIDADE CURRICULAR : PROGRAMAÇÃO ORIENTADA A OBJECTOS



UNIDADE CURRICULAR : PROGRAMAÇÃO ORIENTADA A OBJECTOS

- **Tipos de dados básicos**

Dados básicos	Significado
char	usado para representar os caracteres de 8 bits da tabela ASCII.
int	usado para representar os números inteiros (com 16 ou 32 bits, consoante os sistemas informáticos em que se trabalha.
float	usado para representar os números reais, ou melhor, com partes decimais (com 32 bits).
double	semelhante ao float, mas com uma capacidade dupla (64 bits).
bool	usado para representar valores booleanos ou lógicos de verdadeiro ou falso.
void	tipo especial usado para representar dados vazios ou funções que não retornam valor algum.
Wchar _t	tipo especial de caracteres que permite representar os diferentes caracteres do mundo (16 bits).

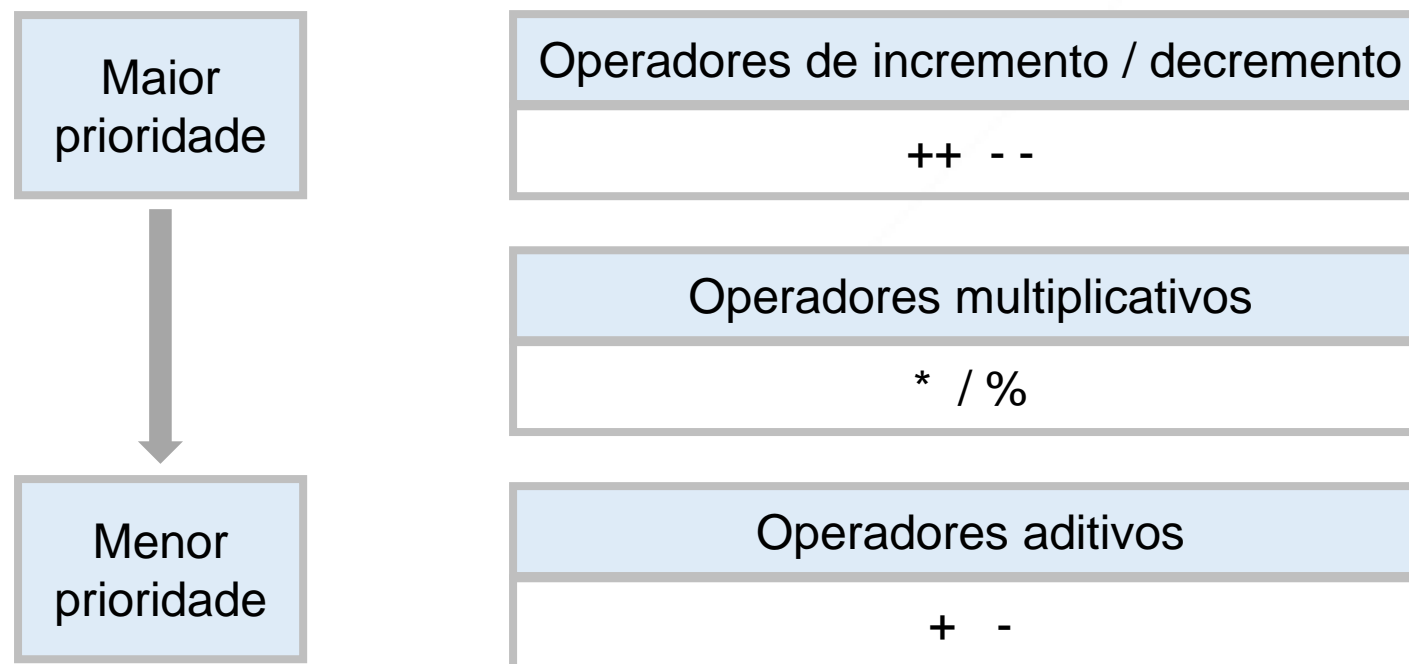
UNIDADE CURRICULAR : PROGRAMAÇÃO ORIENTADA A OBJECTOS

- **Operadores aritméticos**

Operador	Significado
+	Adição
-	Subtracção
*	Multiplicação
/	Divisão
%	Resto da divisão inteira
++	Incremento
--	Decremento

UNIDADE CURRICULAR : PROGRAMAÇÃO ORIENTADA A OBJECTOS

- Operadores aritméticos e Prioridade



UNIDADE CURRICULAR : PROGRAMAÇÃO ORIENTADA A OBJECTOS

- Operadores aritméticos de atribuição

Operadores	Significados
+=	Soma seguida de atribuição (ex.: <code>x+=1;</code> //equivalente a <code>x = x+1;</code>)
-=	Subtracção seguida de atribuição (ex.: <code>x-=2;</code> //equivalente a <code>x= x-2;</code>)
*	Multiplicação seguida de atribuição (ex.: <code>x*=3;</code> //equivalente a <code>x= x*3;</code>)
/=	Divisão seguida de atribuição (ex.: <code>x/=3;</code> //equivalente a <code>x= x/3;</code>)
%=	Resto da divisão inteira seguida de atribuição (ex.: <code>x%=3;</code> //equivalente a <code>x =x%3;</code>)

UNIDADE CURRICULAR : PROGRAMAÇÃO ORIENTADA A OBJECTOS

- **Exemplo**

- Escreva um programa considerando que:

- **int** x =10;
- Instrução: x+=1;
- Output : **11**

- Possível solução:

```
#include <iostream>
using namespace std;
int x = 10;
main(){
    x+=1;
    cout <<x<<" \n";
    system("PAUSE");
}
```



CTeSP

CURSOS TÉCNICOS
SUPERIORES PROFISSIONAIS