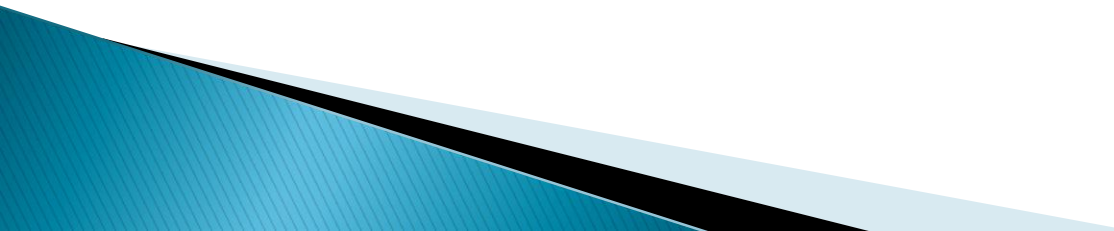


INTRODUÇÃO A LÓGICA DE PROGRAMAÇÃO

Unidade II – Estrutura de um Programa C++

Linguagem de Programação

- ▶ A Linguagem C foi criada por Dennis M. Ritchie e Ken Thompson no laboratório Bell em 1972.
 - ▶ Rapidamente a linguagem C se tornou uma das linguagens mais populares entre os programadores, pois é poderosa, portátil, flexível e foi projetada para ser de propósito geral, estruturada, imperativa, procedural e padronizada.
 - ▶ A linguagem C é muito próxima da linguagem algorítmica (pseudocódigo). Isto ocorre, pois ela é imperativa e estruturada.
- 

A forma de um programa C

- ▶ A linguagem C é baseada no conceito de funções. Um programa C é um conjunto de funções.
- ▶ Uma biblioteca é um arquivo contendo as funções padrão que seu programa pode usar. Essas funções incluem todas as operações de entrada e saída (E/S) e também outras rotinas úteis para realizar as tarefas mais comumente necessárias.

Estrutura Básica de um programa em C

- ▶ A forma geral de um programa em C é a seguinte:

```
< diretivas do pré-processador >
```

```
< declarações globais >;
```

```
main()
```

```
{
```


```
    < declarações locais >; /*
```

```
comentário */
```

```
    < instruções >;
```

```
}
```

```
< outras funções >
```



A função main

- ▶ Um programa em C consiste de uma ou várias funções. A função `main()` deve existir em algum lugar de seu programa e marca o início da execução do programa.

```
main()  ⇐ primeira função a ser executada
{       ⇐ início do corpo da função
}       ⇐ término do corpo da função
```

- ▶ Todas as instruções devem estar dentro das chaves e são executadas na ordem em que as escrevemos. As instruções em C são sempre encerradas por um ponto-e-vírgula (;).

Entrada e Saída em C++

- ▶ A linguagem C++ permite a utilização dos mecanismos de E/S (I/O) definidos em suas bibliotecas.
- ▶ As definições e declarações necessárias para o uso de comandos de entrada e saída estão contidas no arquivo `<iostream.h>`.
- ▶ Nela estão os comando `cin` e `cout` destinados à entrada de dados (via teclado) e saída de dados (via monitor de vídeo).

Comando de Saída

- ▶ Para mostrar os dados em um programa C++, deve-se utilizar o comando `cout`. Por padrão `cout` está associada a uma saída padrão, i.e., ao terminal de vídeo.
- ▶ Sintaxe:

```
cout << expressão;
```
- ▶ Onde:
 - `<<`: é o operador de inserção usado a fim de direcionar a saída de dados para a saída padrão (vídeo).
 - `expressão`: é qualquer combinação de caracteres ou variável.

Comando de Saída

- ▶ Os caracteres que não podem ser obtidos diretamente do teclado para dentro do programa (como a mudança de linha) são escritos em C com a combinação do sinal \ (barra invertida) com outros caracteres, como mostra a tabela abaixo:

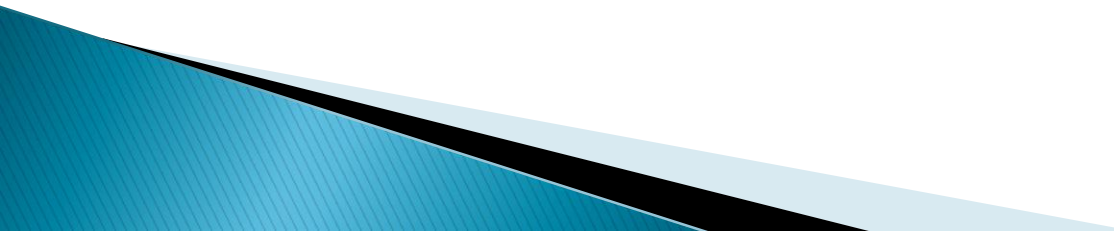
códigos especiais	significado
<code>\n</code>	nova linha
<code>\r</code>	retorno do cursor
<code>\t</code>	tabulação horizontal
<code>\b</code>	retrocesso
<code>\v</code>	tabulação vertical
<code>\a</code>	sinal sonoro
<code>\0</code>	nulo

Comando de Saída

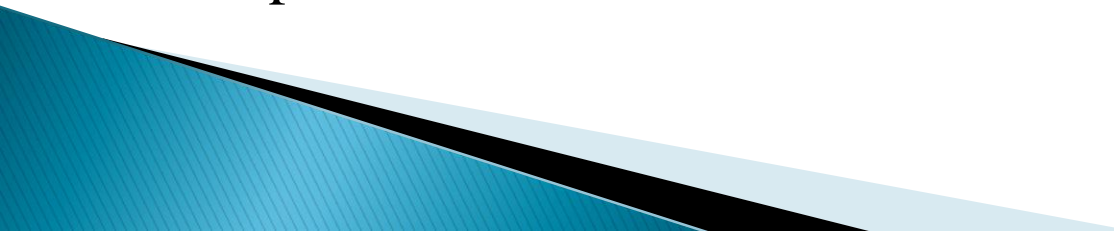
▶ Exemplo:

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Primeiro programa";
    cout << "\n Boa Noite";
}
```

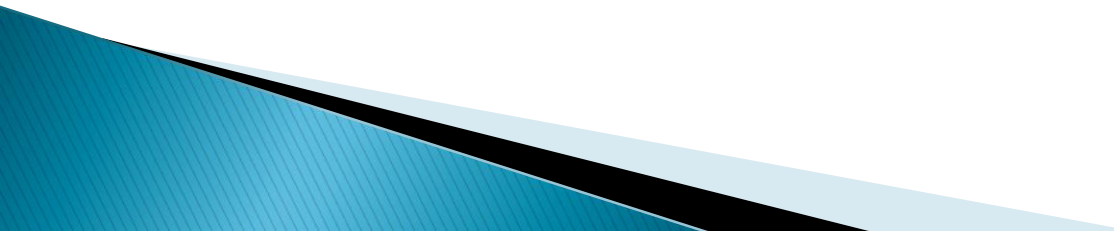
Constantes e Variáveis

- ▶ Na linguagem C, **constantes** referem-se a **valores fixos** que não podem ser alterados pelo programa. O modo como cada constante é representada depende do seu tipo.
 - ▶ As **variáveis** são o aspecto fundamental de qualquer linguagem de programação. Uma variável em C é um espaço de memória (“depósito”) reservado para armazenar um certo tipo de dado e tendo um nome para referenciar o seu conteúdo. Uma variável pode conter, a cada instante, valores diferentes.
- 

Tipos de Variáveis

- ▶ Os **dados** manipulados por um programa podem possuir natureza distinta, isto é, podem ser números, letras, frases, etc. Dependendo da natureza de um dado, algumas operações podem ou não fazer sentido quando aplicadas a eles.
 - ▶ O **tipo** de uma variável define o conjunto de valores ao qual o valor do dado pertence, bem como o conjunto de todas as operações que podem atuar sobre ele.
 - ▶ Em C existem 5 tipos de variáveis básicas: caractere, inteiro, ponto flutuante, ponto flutuante duplo e sem valor.
 - ▶ As palavras reservadas usadas para declarar variáveis desses tipos são: `char`, `int`, `float`, `double` e `void`, respectivamente.
- 

Tipos de Variáveis

- ▶ **int**: inclui os números inteiros e as operações aritméticas e relacionais para estes números. Ex.: 5, 10, 578
 - ▶ **float/double**: inclui os números reais e as operações aritméticas e relacionais para estes números. Ex.: 5.7, 10.67
 - ▶ **char**: inclui os símbolos usados pelo programa (letras, dígitos, símbolos, etc.). As operações definidas para este tipo são igualdade e desigualdade. Os elementos do tipo caractere devem ser escritos, nos algoritmos, entre aspas simples. Ex.: 'a', '@', '5'
- 

Declaração de Variáveis

- ▶ Uma **declaração de variável** é uma instrução para reservar uma quantidade de memória apropriada para armazenar o tipo especificado e indicar que o seu conteúdo será referenciado pelo nome da variável.
- ▶ Uma declaração de variável consiste em um tipo seguido do nome da variável. Exemplo:

```
int num;
```

- ▶ Em C todas as variáveis devem ser declaradas. Se você tiver mais de uma variável do mesmo tipo, poderá declará-las de uma única vez, separando seus nomes por vírgulas.

```
int num1, num2, num3;
```

Comando de Entrada

- ▶ A forma de entrar com dados em um programa C++ é através do comando `cin`. Trata-se de um fluxo associado à entrada padrão do computador, isto é, o teclado.
- ▶ A sintaxe de `cin` é:

```
cin >> variável;
```
- ▶ Onde:
 - `>>`: é o operador de extração usado para direcionar a entrada de dados à entrada padrão (teclado).
 - `variável`: é o nome da variável onde desejamos guardar os valores lidos.
- ▶ O comando `cin` faz com que o programa aguarde que você digite o dado a ser lido e pressione a tecla [ENTER] para finalizar a entrada.

Operador de Atribuição

- ▶ Em C, o sinal de igual (=) não tem a mesma interpretação dada em matemática. Representa a atribuição da expressão à direita ao nome da variável à esquerda.
- ▶ Por exemplo:

```
num = 20;  
letra = 'a';
```
- ▶ Depois da execução do comando a variável `num` passa a armazenar o valor 20 e a variável `letra` passa a armazenar o valor `'a'`.

Operadores Aritméticos

- ▶ O C oferece 5 operadores aritméticos binários (operam sobre dois operandos) e um operador aritmético unário (opera sobre um operando).

operador	operação
+	adição
-	subtração e menos unário
*	multiplicação
/	divisão
%	resto da divisão inteira

Operadores Aritméticos

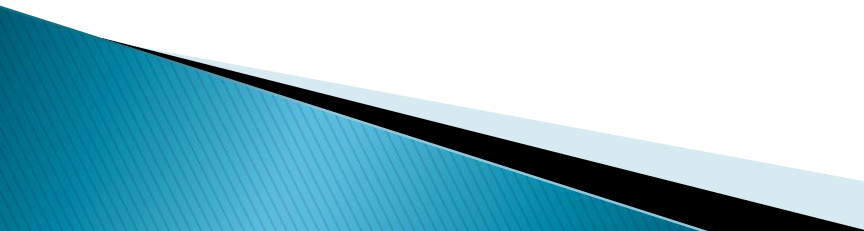
- ▶ Os operadores aritméticos binários representam as operações aritméticas de soma (+), subtração (-), divisão (/) e multiplicação (*) e resto da divisão inteira (%).

- ▶ Exemplo:

```
resultado = num1 + num2 * num3;
```

- ▶ O operador aritmético unário é utilizado somente para indicar a troca do sinal algébrico do valor. Por exemplo:

```
num = -8;
```



Precedência de Operadores

- ▶ Assim como na aritmética padrão, a precedência de operadores nas expressões aritméticas dos programas também é governada pelo uso de parênteses.
- ▶ A menos que os parênteses indiquem o contrário, as operações de divisão e multiplicação são executadas primeiro. Em seguida, temos as operações de adição e subtração.
- ▶ Exemplo:

```
inteiro a, b, c;
```

```
x ← a + b * c;
```

```
x ← (a + b) * c;
```

Um exemplo completo

```
#include <iostream>
using namespace std;
int main()
{
    int num1, num2, resultado;

    cout << "Informe dois numeros: ";
    cin >> num1 >> num2;

    resultado = num1 + num2;

    cout << "\n Soma dos numeros: " << resultado;
}
```

