

# Estrutura Básica

Informática – IFSULDEMINAS

Primeiro Semestre de 2013

# Roteiro

- 1 Variáveis e Tipos de Dados
- 2 Operadores Aritméticos
- 3 Estrutura de um Programa em C

# Variáveis

## Variáveis

Representam regiões da memória às quais associamos um **nome** e um **tipo**. Elas guardam um conteúdo.

- Quando definimos uma variável, definimos também seu nome e seu tipo.
- O tipo de uma variável define qual o tipo de dado que ela poderá armazenar.
- Exemplos:
  - ▶ Variável **nome** do tipo String que representa um nome.
  - ▶ Variável **idade** do tipo inteiro que representa uma idade.

# Variáveis

## Variáveis

Representam regiões da memória às quais associamos um **nome** e um **tipo**. Elas guardam um conteúdo.

- Quando definimos uma variável, definimos também seu nome e seu tipo.
- O tipo de uma variável define qual o tipo de dado que ela poderá armazenar.
- Exemplos:
  - ▶ Variável **nome** do tipo String que representa um nome.
  - ▶ Variável **idade** do tipo inteiro que representa uma idade.

# Variáveis

## Variáveis

Representam regiões da memória às quais associamos um **nome** e um **tipo**. Elas guardam um conteúdo.

- Quando definimos uma variável, definimos também seu nome e seu tipo.
- O tipo de uma variável define qual o tipo de dado que ela poderá armazenar.
- Exemplos:
  - ▶ Variável **nome** do tipo String que representa um nome.
  - ▶ Variável **idade** do tipo inteiro que representa uma idade.

# Variáveis

## Variáveis

Representam regiões da memória às quais associamos um **nome** e um **tipo**. Elas guardam um conteúdo.

- Quando definimos uma variável, definimos também seu nome e seu tipo.
- O tipo de uma variável define qual o tipo de dado que ela poderá armazenar.
- Exemplos:
  - ▶ Variável **nome** do tipo String que representa um nome.
  - ▶ Variável **idade** do tipo inteiro que representa uma idade.

# Tipos em C

A linguagem C tem 5 tipos básicos: **char**, **int**, **float**, **void**, **double**

Tipo	Número de Bits	Intervalo de Valores
char	8 bits	-128 a 127
int	18	-32.768 a 32.767
float	32	3,4E-38 a 3.4E+38
double	64	1,7E-308 a 1,7E+308
void	-	-

- **void** é o tipo vazio ou sem tipo, é utilizado em funções.
- Para cada tipo básico, há modificadores de tipos. Para o tipo **int** temos o **short int**, **long int**, por exemplo.

# Tipos em C

A linguagem C tem 5 tipos básicos: **char**, **int**, **float**, **void**, **double**

Tipo	Número de Bits	Intervalo de Valores
char	8 bits	-128 a 127
int	18	-32.768 a 32.767
float	32	3,4E-38 a 3.4E+38
double	64	1,7E-308 a 1,7E+308
void	-	-

- **void** é o tipo vazio ou sem tipo, é utilizado em funções.
- Para cada tipo básico, há modificadores de tipos. Para o tipo **int** temos o **short int**, **long int**, por exemplo.



# Tipos em C

A linguagem C tem 5 tipos básicos: **char**, **int**, **float**, **void**, **double**

Tipo	Número de Bits	Intervalo de Valores
char	8 bits	-128 a 127
int	18	-32.768 a 32.767
float	32	3,4E-38 a 3.4E+38
double	64	1,7E-308 a 1,7E+308
void	-	-

- **void** é o tipo vazio ou sem tipo, é utilizado em funções.
- Para cada tipo básico, há modificadores de tipos. Para o tipo **int** temos o **short int**, **long int**, por exemplo.

# Tipos de Dados

## Declaração de Variáveis

<tipo> <nome da variável>;

- **int**: representam números inteiros.

## Variáveis Inteiras

```
int ano;  
ano = 2012;
```

- **float**: representam valores numéricos, podendo apresentar casas decimais (separadas por meio do ".").

## Variáveis de Ponto Flutuante

```
float media;  
media = 3.5;
```

# Tipos de Dados

## Declaração de Variáveis

<tipo> <nome da variável>;

- **int**: representam números inteiros.

## Variáveis Inteiras

```
int ano;  
ano = 2012;
```

- **float**: representam valores numéricos, podendo apresentar casas decimais (separadas por meio do ".").

## Variáveis de Ponto Flutuante

```
float media;  
media = 3.5;
```

# Tipos de Dados

- **double**: representam valores numéricos, podendo apresentar casas decimais.

## Variáveis de Ponto Flutuante Duplo

```
float peso;  
peso = 75.8;
```

- **char**: representam valores numéricos inteiros e caracteres de texto. Para indicar um caractere de texto usamos **apóstrofes**.

## Variáveis Caracteres

```
char Ch;  
Ch = 'D';
```

# Tipos de Dados

- **double**: representam valores numéricos, podendo apresentar casas decimais.

## Variáveis de Ponto Flutuante Duplo

```
float peso;  
peso = 75.8;
```

- **char**: representam valores numéricos inteiros e caracteres de texto. Para indicar um caractere de texto usamos **apóstrofes**.

## Variáveis Caracteres

```
char Ch;  
Ch = 'D';
```

# Tipos de Dados

- **String**: representam cadeias de carecteres (palavras ou frases).
- Em C, Strings são vetores de carecteres. Declaração:

## Declaração de Variáveis

```
char <nome da variável>[<tamanho>;
```

- Para indicar uma String usamos **aspas duplas**.

## Variáveis de String

```
char instituto[20];  
instituto = "IFSULDEMINAS";
```

# Nomes de Variáveis

- Devem começar com letras ou underscore "\_".
- Os caracteres subsequentes devem ser letras, números ou underscore.
- Restrições:
  - ▶ Palavras reservadas.
  - ▶ Até 32 caracteres.
- **Obs:** C é uma linguagem case sensitive.

# Nomes de Variáveis

- Devem começar com letras ou underscore "\_".
- Os caracteres subsequentes devem ser letras, números ou underscore.
- Restrições:
  - ▶ Palavras reservadas.
  - ▶ Até 32 caracteres.
- Obs: C é uma linguagem case sensitive.



# Nomes de Variáveis

- Devem começar com letras ou underscore "\_".
- Os caracteres subsequentes devem ser letras, números ou underscore.
- Restrições:
  - ▶ Palavras reservadas.
  - ▶ Até 32 caracteres.
- Obs: C é uma linguagem case sensitive.

# Nomes de Variáveis

- Devem começar com letras ou underscore "\_".
- Os caracteres subsequentes devem ser letras, números ou underscore.
- Restrições:
  - ▶ Palavras reservadas.
  - ▶ Até 32 caracteres.
- **Obs:** C é uma linguagem case sensitive.

# Observações sobre Variáveis

- Devem ser declaradas antes de serem usadas.
- Uma vez que o tipo das variáveis foi definido, este tipo não poderá ser mudado.
- Atribuições de tipos de dados diferentes geram mensagens de erro, detectadas e acusadas pelo compilador.

# Observações sobre Variáveis

- Devem ser declaradas antes de serem usadas.
- Uma vez que o tipo das variáveis foi definido, este tipo não poderá ser mudado.
- Atribuições de tipos de dados diferentes geram mensagens de erro, detectadas e acusadas pelo compilador.

# Observações sobre Variáveis

- Devem ser declaradas antes de serem usadas.
- Uma vez que o tipo das variáveis foi definido, este tipo não poderá ser mudado.
- Atribuições de tipos de dados diferentes geram mensagens de erro, detectadas e acusadas pelo compilador.

# Operadores Aritméticos

Operados	Ação
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Resto de Divisão
++	Incremento
--	Decremento

- O operador / aplicado a inteiros resulta na divisão inteira.
- O operador % deve ser aplicado a números inteiros.

# Operadores Aritméticos

Operados	Ação
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Resto de Divisão
++	Incremento
--	Decremento

- O operador / aplicado a inteiros resulta na divisão inteira.
- O operador % deve ser aplicado a números inteiros.

# Operadores Aritméticos

Operados	Ação
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Resto de Divisão
++	Incremento
--	Decremento

- O operador / aplicado a inteiros resulta na divisão inteira.
- O operador % deve ser aplicado a números inteiros.



# Operadores Aritméticos

## Operadores Aritméticos

```
int numero1, numero2, resultado1, resultado2;  
float div1, div2, numero3;  
numero1 = 13;  
numero2 = 5;  
numero3 = 6;  
resultado1 = numero1/numero2;  
resultado2 = numero1%numero2;  
div1 = numero3/numero2;  
div2 = numero1/numero2;
```

Qual o valor das variáveis **resultado1**, **resultado2**, **div1** e **div2**?

# Operadores Aritméticos

## Operadores Aritméticos

```
int numero1, numero2, resultado1, resultado2;  
float div1, div2, numero3;  
numero1 = 13;  
numero2 = 5;  
numero3 = 6;  
resultado1 = numero1/numero2;  
resultado2 = numero1%numero2;  
div1 = numero3/numero2;  
div2 = numero1/numero2;
```

Qual o valor das variáveis **resultado1**, **resultado2**, **div1** e **div2**?

# Operadores Aritméticos ++ e --

## Operadores ++ e --

- Os operadores alteram o valor de uma variável de 1. Então:

`x++;`

`y--;`

são equivalentes a:

`x = x + 1;`

`y = y - 1;`

- Os operadores podem ser pré-fixados ou pós-fixados: `++x` ou `x++`
- Pré-fixados:** incrementam ou decrementam e retornam o valor da variável.
- Pós-fixados:** retornam o valor da variável e depois incrementam ou decrementam.

# Operadores Aritméticos ++ e --

## Operadores ++ e --

- Os operadores alteram o valor de uma variável de 1. Então:

`x++;`

`y--;`

são equivalentes a:

`x = x + 1;`

`y = y - 1;`

- Os operadores podem ser pré-fixados ou pós-fixados: `++x` ou `x++`
- Pré-fixados:** incrementam ou decrementam e retornam o valor da variável.
- Pós-fixados:** retornam o valor da variável e depois incrementam ou decrementam.

# Operadores Aritméticos ++ e --

## Operadores ++ e --

- Os operadores alteram o valor de uma variável de 1. Então:

`x++;`

`y--;`

são equivalentes a:

`x = x + 1;`

`y = y - 1;`

- Os operadores podem ser pré-fixados ou pós-fixados: `++x` ou `x++`
- Pré-fixados:** incrementam ou decrementam e retornam o valor da variável.
- Pós-fixados:** retornam o valor da variável e depois incrementam ou decrementam.

# Operadores Aritméticos ++ e --

## Operadores ++ e --

- Os operadores alteram o valor de uma variável de 1. Então:

`x++;`

`y--;`

são equivalentes a:

`x = x + 1;`

`y = y - 1;`

- Os operadores podem ser pré-fixados ou pós-fixados: `++x` ou `x++`
- Pré-fixados:** incrementam ou decrementam e retornam o valor da variável.
- Pós-fixados:** retornam o valor da variável e depois incrementam ou decrementam.

# Operadores Aritméticos

## Incremento e Decremento

```
int a, b, c, d, e;  
a = 10;  
b = ++a;  
c = a++;  
d = -a;  
e = a--;
```

Qual o valor das variáveis **b**, **c**, **d** e **e**?

# Operadores Aritméticos

## Incremento e Decremento

```
int a, b, c, d, e;  
a = 10;  
b = ++a;  
c = a++;  
d = -a;  
e = a--;
```

Qual o valor das variáveis **b**, **c**, **d** e **e**?



# Estrutura Básica de um Programa em C

## Estrutura básica de um programa C

*Declaração de bibliotecas Usadas*

*Declaração de variáveis*

```
int main(){
```

*Declaração de variáveis*

*Comandos*

...

*Comandos*

```
}
```

# Estrutura Básica de um Programa em C

## Exemplo

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    int a;
    int b, c;

    a = 3 + 23;
    b = a + 18;
    c = b - a;

    system("PAUSE");
}
```