

# Linguagem C: Declarações, Variáveis e Comandos de Atribuição

Prof. Paulo R. S. L. Coelho

paulo@facom.ufu.br

Faculdade de Computação  
Universidade Federal de Uberlândia



# Organização

- 1 Declarações
- 2 Variáveis
- 3 Comandos de Atribuição
  - Definição
  - Operadores Matemáticos
- 4 Exercícios



# Organização

- 1 Declarações
- 2 Variáveis
- 3 Comandos de Atribuição
  - Definição
  - Operadores Matemáticos
- 4 Exercícios



# Declarações

- Nomes de variáveis devem respeitar convenção para identificadores (aula de introdução).
- Palavras reservadas não podem ser usadas como nomes de variáveis.
- Deve ter um tipo atribuído em sua declaração;
- Ex:

```
(...)  
int nota;  
double media;  
char _caractere;  
(...)
```



# Constantes Simbólicas

- Criadas com diretivas `define`.
- Pré-processador substitui constante simbólica pelo valor.
- Ex:

```
#include <stdio.h>

#define PI 3.14159

int main() {
    double raio = 0.0, area;
    printf("Digite valor do raio\n");
    scanf("%f", &raio);
    area = PI * raio * raio;
    printf("A Area do circulo de raio %f eh %f", raio, area);
    return 0;
}
```



# Declaração de Constantes

- Uma constante pode ser declarada adicionando-se o modificador `const` à declaração de uma variável.
- Ex:

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

```
int main() {  
    const double PI = 3.14159;  
    double raio = 0.0, area;  
    printf("Digite valor do raio\n");  
    scanf("%f", &raio);  
    area = PI * raio * raio;  
    printf("A Area do circulo de raio %f eh %f", raio, area);  
    return 0;  
}
```



# Organização

- 1 Declarações
- 2 Variáveis
- 3 Comandos de Atribuição
  - Definição
  - Operadores Matemáticos
- 4 Exercícios



# Variáveis de Tipo Primitivo I

- Cada linguagem fornece ao programador uma relação de tipos padronizados para suas variáveis.
- Estes tipos recebem o nome de **tipos primitivos**.
- A linguagem C fornece tipos primitivos para variáveis inteiras, reais e de tipo caractere.
- É possível também a criação de novos tipos.
- Uma declaração de variáveis tem a seguinte forma:  
`tipo variavel1, variavel2, ..., variavelN;`
- **Variáveis inteiras:** `(unsigned) int`, `(unsigned) short`, `(unsigned) long`
- **Variáveis reais:** `float`, `double`
- **Variáveis caracteres:** `(unsigned) char`



# Organização

- 1 Declarações
- 2 Variáveis
- 3 Comandos de Atribuição**
  - Definição
  - Operadores Matemáticos
- 4 Exercícios



# Atribuição

- A forma geral de um comando de atribuição é:  
`variavel = expressao;`
- A execução do comando consiste em calcular o valor da expressão e atribuí-lo à referida variável;
- Esta expressão pode conter constantes, variáveis, operadores, e chamadas de funções.
- Seu resultado deve ser um único valor, de um determinado tipo (inteiro, real, caractere, lógico, ... ).



# Introdução

- Estes operadores permitem a execução de operações aritméticas pelo computador, atualizando o valor de variáveis e realizando o processamento necessário para execução do algoritmo.
- Os principais operadores são  $*$ ,  $/$ ,  $+$  e  $-$  além do operador  $\%$ , que indica o resto de uma divisão inteira.
- Ex:

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

```
int main() {  
    int n1, n2;  
    printf("Entre um número:");  
    scanf("%d", &n1);  
    n2 = n1 * 2;  
    printf("o dobro do número %d é: %d", n1, n2);  
    return 0;  
}
```



# Construções especiais

- As operações básicas permitem a construções de algumas expressões especiais em C:

- `i += 2;` //equivale a `i = i + 2;`
- `i -= j + 1;` //equivale a `i = i - (j + 1);`
- `i *= 4;` //equivale a `i = i * 4;`

- Ex:

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

```
int main() {  
    int n1, n2, n3;  
    printf("Entre um número:");  
    scanf("%d", &n1);  
    printf("Entre outro número:");  
    scanf("%d", &n2);  
    n3 = n2;  
    n3 -= n1; // equivale a n3 = n2 - n1  
    printf("a diferença entre %d e %d é: %d", n1, n2, n3);  
    return 0;  
}
```



# Incremento e Decremento

- Existem ainda duas operações especiais:
  - Incremento (++): aumenta de um o valor da variável;
  - Decremento (--): diminui de um o valor da variável;
- Dependendo da posição do operador o resultado pode variar;
- Ex:

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

```
int main() {  
    int n1;  
    printf("Entre um número:");  
    scanf("%d", &n1);  
    printf("%d", n1++); //n1 é incr. após impressão  
    printf("%d", n1);  
    printf("%d", ++n1); //n1 é incr. antes  
    printf("%d", n1);  
    return 0;  
}
```



# Precedência

- O nível de precedência dos operadores é avaliado da esquerda para a direita.
- Os parênteses podem ser utilizados para alterar a ordem da avaliação.

1 ++ --

2 \* / %

3 + -

- Ex:

```
#include <stdio.h>

int main() {
    int n1, n2, n3;
    printf("Entre um número:");
    scanf("%d", &n1);
    printf("Entre outro número:");
    scanf("%d", &n2);
    n3 = n2++ + ++n1 % 3 * 2 + 7;
    printf("Números: %d, %d e %d", n1, n2, n3);
    return 0;
}
```

# Organização

- 1 Declarações
- 2 Variáveis
- 3 Comandos de Atribuição
  - Definição
  - Operadores Matemáticos
- 4 Exercícios



# Exercícios I

- 1 Diga o que será escrito na tela durante a execução do seguinte trecho de código:

```
int a, b = 0, c = 0;
a = ++b + ++c;
printf("%d %d %d\n", a, b, c);
a = b++ + c++;
printf("%d %d %d\n", a, b, c);
a = ++b + c++;
printf("%d %d %d\n", a, b, c);
a = b-- + --c;
printf("%d %d %d\n", a, b, c);
```

- 2 Escreva um programa em C que leia uma temperatura em graus celsius (do tipo `double`) e retorne a temperatura equivalente em fahrenheit;

Dado:  $F = \frac{9C}{5} + 32$



## Exercícios II

- 3 Escreva um programa em C que leia uma temperatura em fahrenheit (do tipo `double`) e retorne a temperatura equivalente em celsius;

$$\text{Dado: } C = \frac{(F-32) \times 5}{9}$$

- 4 Escreva um programa em C que leia o peso e a altura de uma pessoa e calcule o seu IMC, bem como os valores máximo e mínimo do peso, segundo a OMS, considerados saudáveis.

$$\text{Dado: } IMC = \frac{\text{Peso}}{(\text{Altura})^2}, IMC_{min} = 18.5, IMC_{max} = 25$$



# Respostas I

1 > 2 1 1  
> 2 2 2  
> 5 3 3  
> 5 2 2

2 #include <stdio.h>

```
int main() {  
    double cels, faren;  
    printf("Entre com a temperatura em graus celsius\n");  
    scanf("%f", &cels); //leitura do valor  
    faren = 9*cels/5 + 32; //calculo do resultado  
    printf("%f graus celsius sao %f graus fahrenheit\n",  
           cels, faren);  
    return 0;  
}
```



## Respostas II

```
8 #include <stdio.h>

int main() {
    double cels, faren;
    printf("Entre com a temperatura em fahrenheit\n");
    scanf("%f", &faren); //leitura do valor
    cels = ((faren - 32)*5)/9; //calculo do resultado
    printf("%f graus fahrenheit sao %f graus celsius\n",
           faren, cels);

    return 0;
}
```



## Respostas III

```
4 #include <stdio.h>
#include <math.h>

int main() {
    double peso, altura, imc, pesoMax, pesoMin;

    printf("Entre com peso:\n");
    scanf("%f", &peso);
    printf("Entre com altura:\n");
    scanf("%f", &altura);

    imc = peso / pow(altura, 2);    // pow(a,b) -> a^b
    printf("\nO valor do IMC é %f\n\n", imc);

    pesoMax = 25 * pow(altura, 2);
    pesoMin = 18.5 * pow(altura, 2);
    printf("O menor peso para uma altura de %f é %f\n",
           altura, pesoMin);
    printf("O maior peso para uma altura de %f é %f\n",
```



# Respostas IV

```
}  
    return 0;  
}
```

```
altura, pesoMax);
```

