

# Curso de C

## *Estruturas da Linguagem C*

# Estruturas de Linguagem C

## Objetivos:

- Compreender:
  - As componentes que compõem um programa em C

# Estruturas de Linguagem C

## Roteiro:

- Blocos
- Sentenças
- Expressões

# Estruturas de Linguagem

## Um programa C é composto por:

- Expressões

Ex.: `media = (nota_a + 2 * nota_b) / 3;`

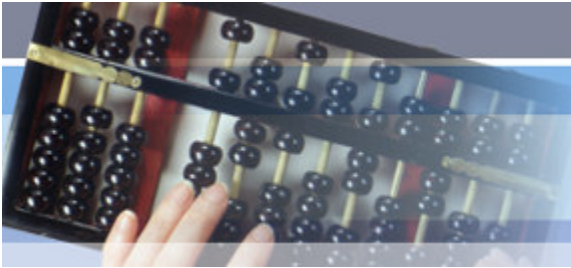
- Sentenças

Ex.: `int variavel = 3;`

- Blocos

Ex.: 

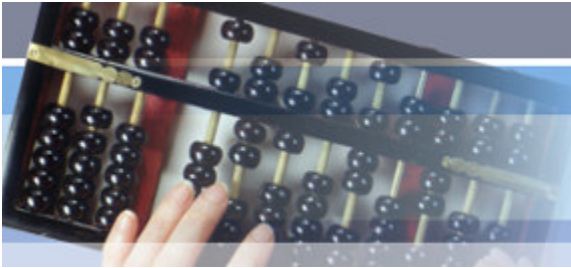
```
if (media >= 5.0)
{
    printf("Aprovado!");
    ...
};
```



# Sentenças

## O que são *sentenças*?

- Unidade completa.
- São um passo do algoritmo ou uma declaração.
- Terminam em ponto-e-vírgula.
- Exemplos:
  - Declarações
  - Conjunto de expressões (ex: atribuição e operadores)
  - Operação de leitura ou escrita (ex: `printf`, `scanf`)
  - Controle de fluxo de execução



# Sentenças

## Exemplo de Sentenças:

```
float nota1, nota2;  
float media;  
printf("Digite as duas notas: ");  
scanf("%f %f", &nota1, &nota2);  
media = (nota1 + nota2) / 2;  
printf("Média: %f", media);  
return 0;
```

O programa é uma seqüência de sentenças.

Cada uma é um passo do algoritmo ou  
uma declaração.



# Expressões

## O que é uma *Expressão*?

- Unidade indivisível de código C
- Define:
  - Operação
  - Valores utilizados
  - Propagação do resultado
- Muitas expressões são parecidas com expressões matemáticas.

# Expressões

## Cuidado:

- Pensamento matemático:

$b \leftarrow 3$

$a = b$

...

$b \leftarrow 5$

Propaga para  
expressões  
anteriores

$\Rightarrow a = b = 5$

Expressões representam afirmações verdadeiras.

- Pensamento em Código C:

$b = 3;$

$a = b;$

...

$b = 5;$

Não afeta  
sentenças  
anteriores

$\Rightarrow b = 5$ , mas  $a = 3$

Expressões representam operações independentes



# Expressões

## Expressões e Sentenças:

b = 3;

a = b;

...

b = 5;

Não afeta  
sentenças  
anteriores

- Sentenças contém expressões independentes.
- Uma expressão é avaliada somente no momento da execução.



# Blocos

## O que são *Blocos*?

- Conjunto de sentenças
- Agrupadas entre chaves { ... }
- Define lógica de execução seqüencial.
- Sentenças do mesmo bloco são executadas seqüencialmente

# Blocos

## Exemplo de Bloco:

```
int main(int argc, char* argv[]) {  
    float nota1, nota2;  
    float media;  
    printf("Digite as duas notas: ");  
    scanf("%f %f", &nota1, &nota2);  
    media = (nota1 + nota2) / 2;  
    printf("Média: %f", media);  
    return 0;  
}
```

# Estruturas de Linguagem

*Fim do Capítulo*