PONTIFÍCIA UNIVERSIDADE CATÓLICA DE SÃO PAULO
Ciência da Computação

Laboratório de Programação 01

Principais Construções da Linguagem C

Baseado no capítulo 1 do livro "C a linguagem de programação padrão ANSI" de Kernigan e Ritchie

Prof. Carlos Eduardo de B. Paes
Departamento de Computação
Pontifícia Universidade Católica de São Paulo
carlosp@pucsp.br

Objetivos

Estudar as principais construções que compõem os elementos básicos da Linguagem C

Principais Construções

- **Estrutura básica de um programa C**
- **Declaração de variáveis**
- **Mecanismo de atribuição**
- **Operações aritméticas**
- **Operadores relacionais e lógicos**
- **Algumas estruturas de controle básicas**

Estrutura Básica de um Programa C

❑ **Um programa em C basicamente consiste de *declarações* e *comandos***

❑ **Os *comandos* especificam as operações de computação a serem feitas**

❑ **As *declarações* descrevem *variáveis que* armazenam valores usados durante a computação**

Estrutura Básica de um Programa C

❑ **Todo o programa em C deverá conter um "ponto" de início da execução do programa.**
❑ **Este "ponto" é conhecido como "*main()*"**
❑ **O "main()" contém um escopo de definição/agrupamento de comandos na linguagem**
❑ **Este escopo é definido pelos símbolos de chaves { e } → inicio e fim, respectivamente.**
❑ **Antes deste ponto de entrada, normalmente especifica-se a inclusão de um arquivo contendo alguns serviços de entrada e saída**
❑ **Importante: vamos estudar mais tarde o que significa este "ponto" de inicio da execução e quais são os serviços de entrada e saída**

Estrutura Básica de um Programa C

Nosso primeiro programa em C

```
#include <stdio.h>

main( ) {

    printf("primeiro programa em C\n");

}
```

Estrutura Básica de um Programa C

Análise do nosso primeiro programa em C

- **A primeira linha do programa, #include <stdio.h>, diz ao compilador para incluir informações sobre a biblioteca-padrão de entrada e saída**
- **Os parêntese após o "main" delimitam a lista de argumentos. No exemplo, não existem argumentos**
- **Os comandos de uma função são delimitados por chaves { e }**

Estrutura Básica de um Programa C

Análise do nosso primeiro programa em C

- **Dentro do "main()" foi implementado apenas um comando.**

  ***printf*("primeiro programa \n");**

- **Este comando corresponde a um serviço implementado por uma biblioteca-padrão**

Declaração de variáveis em C

- **Alguns tipos de dados pré-definidos pela linguagem C:**
  - ***char* : tipo caractere de um byte**
  - ***int* : tipo inteiro**
  - ***float* : tipo ponto flutuante**
  - ***void* : corresponde a nada (nenhum tipo)**

- **Por exemplo:**

```
int i, j, k;
float x, y, z;
char ch;
```

Mecanismo de atribuição

Sintaxe

<comando_atribuição> → <identificador> = <expressão>

Semântica

Atribui um valor a uma variável com o nome definido por um identificador.

Exemplo:

```
void main() {

    float soma;
    int contador;
    char ch;

    soma = 0.0;
    contador = 0;
    ch = 'A';
}
```

Operações Aritméticas

Sintaxe da declaração

<comando_atribuição> → <identificador> = <expressão>

<expressão> → <identificador> [<operador> <identificador>] |
<constante>

<operador> → (+,-,\,*,%)

Semântica da declaração

Realiza uma operação aritmética retornado o resultado para o operando definido do lado esquerdo da atribuição

Operações Aritméticas

Exemplo

```
void main(void)
{
 int x, y, w;

 z = z + 1;
 x = (z - w) % 100;
 w = w - 1;
}
```

Operações Aritméticas

Exercício 1

Com base na linguagem C, verifique se o programa apresentado no exemplo anterior está sintaticamente correto.

Operadores Relacionais e Lógicos

Sintaxe

<expressão> → <identificador> [<operador> <identificador>] |
<constante>

<operador> → (&&,||,<,>,==,>=,<=,!=)

Semântica

- **Operador relacional → realiza a comparação entre dois operandos**
- **Operadores Lógicos → servem para combinar resultados de comparações e são geralmente utilizados nas instruções condicionais.**
- **Na linguagem C não existe o tipo primitivo booleano, ou seja, que permite a uma variável assumir o valor verdadeiro ou falso. O valor zero (0) é falso; qualquer valor diferente de zero é verdadeiro e é representado pelo inteiro um (1).**

Operadores Relacionais e Lógicos

Operadores relacionais:

- **> maior**
- **>= maior ou igual**
- **< menor**
- **<= menor ou igual**
- **== igual**
- **!= deferente**

Operadores Relacionais e Lógicos

Exemplo:

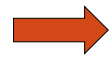
```
#include <sddio.h>

void main( ) {

   int verdadeiro,falso;

   verdadeiro = (15 < 20);
   falso = (15 == 20);

   printf("Resultado1 = %d\n",verdadeiro);
   printf("Resultado2 = %d\n",falso);
}
```



```
Resultado1 = 1
Resultado2 = 0
```

Algumas Estruturas de Controle Básicas

Comando IF - ELSE

Sintaxe

<**comando_seleção**> → if (<expressão>) <comando> [else <comando>]

Semântica

- Comando usado para expressar decisões
- A parte *else* é opcional
- A <*expressão*> é avaliada; se for verdadeira (ou seja, se expressão tiver um valor diferente de zero), o <*comando*> é executado.

Algumas Estruturas de Controle Básicas

Exemplo IF

Considere o seguinte exemplo do comando IF:

```
#include <stdio.h>

void main() {

    int A,B,C;

    printf("Entre com o valor de A,B e C:);
    scanf("%d%d%d",&A,&B,&C);
    if (A>B && A>C) printf("\nA é o maior !");
    if (B>A && B>C) printf("\nB é o maior !");
    if (C>A && C>B) printf("\nC é o maior !");
}
```

```
Entre com o valor A,B e C: 10 30 12

B é o maior !
```

Algumas Estruturas de Controle Básicas

Exemplo IF-ELSE

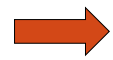
Considere o seguinte exemplo do comando IF:

```
#include <stdio.h>

void main() {

    int A,B,C;

    printf("Entre com o valor de A,B e C:);
    scanf("%d%d%d",&A,&B,&C);
    if (A>B && A>C) printf("\nA é o maior !");
      else
        if (B>C) printf("\nB é o maior !");
          else
            printf("\nC é o maior !");
}
```



```
Entre com o valor A,B e C: 5 50 89

C é o maior !
```

Algumas Estruturas de Controle Básicas

Exercício 2

Analise o seguinte programa implementando na linguagem C

```
#include <stdio.h>

void main() {

    int anos;

    printf("Quantos anos você tem?\n");
    scanf(&ano);
    if (ano < 30)
        printf("Você é muito jovem \n");
}
```

Algumas Estruturas de Controle Básicas

Comando WHILE

Sintaxe

<comando_iteração> → while "(" <expressão> ")" <comando>

Semântica

- **A <expressão> é avaliada. Se for verdadeira (valor diferente de zero), <comando> (corpo do laço) é executado. Então a condição é retestada, e se verdadeira, o corpo <comando> é executado novamente.**

Algumas Estruturas de Controle Básicas

Exemplo do Comando WHILE

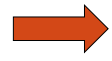
```
#include <stdio.h>

void main () {

    int i = 0;

     while( i < 20) {

          printf("Valor de i = %d\n",i);
          i = i+1;
     }
}
```



```
Valor de i = 1
Valor de i = 2
Valor de i = 3
Valor de i = 4
Valor de i = 5
Valor de i = 6
Valor de i = 7
Valor de i = 8
Valor de i = 9
Valor de i = 10
Valor de i = 11
Valor de i = 12
Valor de i = 13
Valor de i = 14
Valor de i = 15
Valor de i = 16
Valor de i = 17
Valor de i = 18
Valor de i = 19
```

Entrada e Saída em C

printf e scanf

- Os comandos (serviços da biblioteca padrão) *printf()* e *scanf()* permitem escrever no vídeo e ler do teclado, respectivamente, o valor de variáveis
- Estes "comandos" têm como primeiro parâmetro uma string especificando o formato e a ordem das variáveis a escrever ou a ler
- Seguem-se como parâmetros às próprias variáveis pela ordem especificada.
- Na string de formatação indica-se o local e o tipo de um valor de variável através do caractere % seguido de uma letra indicadora do tipo.

Entrada e Saída em C

printf e scanf

- Alguns dos tipos suportados são:
  - ✓ %c - char
  - ✓ %d - int's
  - ✓ %f - float's
- Um exemplo:

```
printf("Os valores das três variáveis são: %c, %d, %f\n", ch, i, x);
```

Entrada e Saída em C

printf e scanf

- A modificação de formatos pode ocorrer para especificar largura e número de casas decimais. Assim o modificador é colocado entre o sinal % e o código do formato. Se tivermos %10f informa que o campo terá 10 posições incluindo a parte inteira o ponto e a parte decimal. Se tivermos %12.3f informa que terá 12 posições no total com 3 casas decimais.

```
#include <stdio.h>

void main(void) {

      double item;
      item = 10.12304;

      printf("%f\n", item);
      printf("%5.2f\n", item);

}
```

Resultado produzido

```
10.123040
10.12
```

Entrada e Saída em C

printf e scanf

❑ As cadeias de caracteres em C definem-se entre aspas ". . .", os caracteres simples aparecem entre apostrofes '.'; o texto normal da string de formatação aparece de mesma forma no vídeo, os valores das variáveis aparecem nos locais indicados pelo caractere %; seguem-se as próprias variáveis que deverão aparecer pela mesma ordem e com os tipos indicados na string de formatação.

❑ O comando scanf() lê valores do teclado para variáveis. A sua estrutura é semelhante a printf().

Por exemplo:

```
scanf("%c%d%f", &ch, &i, &x);
```