

SSC0101 - ICC1 – Teórica

---

Introdução à Ciência da Computação I

**Tipos de Dados Simples  
(Continuação)**

Prof. Vanderlei Bonato: [vbonato@icmc.usp.br](mailto:vbonato@icmc.usp.br)

Prof. Claudio Fabiano Motta Toledo: [claudio@icmc.usp.br](mailto:claudio@icmc.usp.br)

---

# Sumário

---

- Variáveis
- Declaração de Variáveis
- Comandos de atribuição
- Operadores
- Conversão de Tipo na Linguagem C

# Variáveis

---

- O que são variáveis?
  - São referências a áreas de memória do computador que armazenam dados de interesse do programador, dados esses que podem ser alterados a qualquer momento
- A declaração de variáveis é definida pelo programador de acordo com a necessidade para a resolução do problema
- Cada variável é organizada de forma a se garantir a integridade dos dados que contém conforme o tipo definido para ela

# Declaração de variáveis - Pseudocódigo

---

- Ordem: <variable-names> <type-specifier>

- Exemplo:

```
DECLARE  
  x, y NUMÉRICO  
  sexo LITERAL  
  teste LÓGICO
```

# Declaração de variáveis – Linguagem C

---

- Ordem: <type-specifier> <variable-names>
- Nome pode ter até 32 caracteres
- Devem começar com letra ou sublinhado (\_), sendo que os caracteres subsequentes devem ser letras, números ou sublinhado (\_)
- Não podem coincidir com nomes de palavras reservadas, nem de funções declaradas pelo programador ou em bibliotecas do C

# Declaração de variáveis – Linguagem C

---

- C é "case sensitive", ou seja, maiúsculas são distintas de minúsculas.

Nome  $\neq$  nome  $\neq$  NOME  $\neq$  NoMe

- Exemplo:

```
int x;
```

```
int x, y;
```

```
char sexo;
```

```
char nome[40];
```

# Comandos de atribuição - Pseudocódigo

---

- Para atribuir um conteúdo a uma variável, utiliza-se a seguinte instrução:

<variável> ← <conteúdo>

- O conteúdo atribuído à variável pode ser uma literal (número ou caractere) ou outra variável, e seu tipo deve ser compatível com o tipo da variável

$x \leftarrow 4$

$x \leftarrow x+2$

$y \leftarrow 2.5$

sexo ← "F"

teste ← falso

# Comandos de atribuição - Linguagem C

---

- Para atribuir um conteúdo a uma variável, utiliza-se a seguinte instrução (note o sinal de igualdade):

`<variável> = <conteúdo>`

- O conteúdo atribuído à variável pode ser uma literal (número ou caractere) ou outra variável, e seu tipo deve ser compatível com o tipo da variável

- Algumas variações nesse formato são admitidas  
Por exemplo:

`<variável1> = <variável2> = ... = <conteúdo>;`

# Comandos de atribuição - Linguagem C

---

- Exemplo de atribuições de valores ou operações a variáveis (sinal de igualdade)

```
x = 4;
```

```
x = x + 2;
```

```
y = 2.5;
```

```
sexo = 'F';
```

- Em C um caractere é representado entre apóstrofos e uma cadeia de caracteres entre aspas
- Para armazenar uma cadeia de caracteres numa variável deve-se utilizar uma função para manipulação dos mesmos, como:

```
strcpy(nome, "Joao");
```

# Comandos de atribuição - Linguagem C

---

- Variável Global x Variável Local
  - A variável global é visível em todo o programa, enquanto que a local é visível somente dentro da função onde foi declarada

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

```
int x=1;           // x é uma variavel declarada globalmente
```

```
int main()
```

```
{
```

```
    int y;           // y é uma variavel declarada localmente
```

```
    y = 2 * x;
```

```
}
```

# Operadores - Matemáticos

---

- Principais operadores matemáticos

Operação	Operador
Adição	+
Subtração	-
Multiplicação	*
Divisão	/
Resto da Divisão	%
Incremento (+1)	++
Decremento (-1)	--
Sinal Negativo	-

- O operador % é utilizado apenas com inteiros.

# Operadores matemáticos e de atribuição

Operador	Exemplo	Comentário
<code>+=</code>	<code>x += y</code>	Equivale a $X = X + Y$ .
<code>-=</code>	<code>x -= y</code>	Equivale a $X = X - Y$ .
<code>*=</code>	<code>x *= y</code>	Equivale a $X = X * Y$ .
<code>/=</code>	<code>x /= y</code>	Equivale a $X = X / Y$ .
<code>%=</code>	<code>x %= y</code>	Equivale a $X = X \% Y$ .
<code>++</code>	<code>x++</code>	Equivale a $X = X + 1$ .
<code>++</code>	<code>y = ++x</code>	Equivale a $X = X + 1$ e depois $Y = X$ .
<code>++</code>	<code>y = x++</code>	Equivale a $Y = X$ e depois $X = X + 1$ .
<code>--</code>	<code>x--</code>	Equivale a $X = X - 1$ .
<code>--</code>	<code>y = --x</code>	Equivale a $X = X - 1$ e depois $Y = X$ .
<code>--</code>	<code>y = x--</code>	Equivale a $Y = X$ e depois $X = X - 1$ .

# Exemplos de expressões válidas

---

int x, y, z;

x=y=z=15; /\* ⇔ x=15; y=15; z=15; \*/

x=x+2; /\* ⇔ x passa a valer seu valor anterior + 2 \*/

x++; /\* ⇔ x=x+1; \*/

x=++y; /\* ⇔ y=y+1; x=y; \*/

x=y++; /\* ⇔ x=y; y=y+1; \*/

x=-x; /\* ⇔ x = (-1).x \*/

x=x+y-(z--); /\* ⇔ x=x+y-z; z=z-1; \*/

x+=y; /\* ⇔ x=x+y; \*/

x=(++y,y+2); /\* ⇔ y=y+1; x=y+2;\*/

x=(y++,y+2); /\* ⇔ y=y+1; x=y+2;\*/

# Operadores - Relacionais

---

Operador	Exemplo	Comentário
==	$x == y$	Conteúdo de x é igual ao conteúdo de y
!=	$x != y$	Conteúdo de x é diferente do conteúdo de y
<=	$x <= y$	Conteúdo de x é menor ou igual ao conteúdo de y
>=	$x >= y$	Conteúdo de x é maior ou igual ao conteúdo de y
<	$x < y$	Conteúdo de x é menor que o conteúdo de y
>	$x > y$	Conteúdo de x é maior que o conteúdo de y

# Operadores - Relacionais

---

- O resultado retornado por esses operadores será verdadeiro (1) ou falso (0).
- $>$ ,  $>=$ ,  $<$ ,  $<=$  têm a mesma precedência
- Os operadores de igualdade  $==$  e  $!=$  tem precedência sobre os demais.
- Os operadores relacionais possuem precedência menor que os operadores aritméticos.
- Exemplo :  $b < a-1 \Leftrightarrow b < (a-1)$

# Operadores - Lógicos

---

Operador	Exemplo	Comentário
&&	x && y	x e y (x and y)
	x   y	x ou y ( x or y)
!	!x	não x (not x)

x	y	x && y	x    y	! x	!y
V	V	V	V	F	F
V	F	F	V	F	V
F	V	F	V	V	F
F	F	F	F	V	V

# Operadores - Lógicos

---

- Expressões são avaliadas da esquerda para a direita.
- A avaliação termina quando o resultado lógico for conhecido.
- Exemplo:

```
int x = 2, y=3;
```

```
If ((x>=2) || (y>3) &&(2*y<4)) ...
```

```
( V or F ) && F
```

```
V && F  $\Rightarrow$  F
```

# Operadores - Lógicos bit-a-bit

Operador	Comentário
&	e (and)
	ou inclusivo (OR)
^	ou exclusivo (XOR)
<<	deslocamento de bits à esquerda
>>	deslocamento de bits à direita
~	complemento de um (not)

x	y	x & y	x   y	x ^ y	~x
1	1	1	1	0	0
1	0	0	1	1	0
0	1	0	1	1	1
0	0	0	0	0	1

# Operadores - Lógicos bit-a-bit

---

- O operador & é utilizado para mascarar algum conjunto de bits.
  - Exemplo:  $n = n \& 255$ , onde  $255 = 2^8 - 1$

255: 00000000 00000000 00000000 11111111

Teremos apenas os valores do byte de menor ordem armazenados em n.

# Operadores - Lógicos bit-a-bit

---

- O operador  $|$  é utilizado para ativar bits

- Exemplo:  $x = x | \text{ATIVO}$

$x$  passa a ter como 1 os mesmos bits que são 1 em ATIVO.

- O operador  $\wedge$  torna 1 em cada posição onde os operandos possuem bits diferentes

# Operadores - Lógicos bit-a-bit

---

unsigned char x		
Operação	representação	Valor
x=13	00001101	13
x<<1	00011010	26
x<<3	11010000	208
x>>1	01101000	104
x>>4	00000110	6

- variável >> qtd\_deslocamentos\_direita.
- Um deslocando à direita divide a representação por 2.
- variável << qtd\_deslocamentos\_esquerda.
- Um deslocando uma posição à esquerda multiplica a representação por 2.
- Não há rotação, bits deslocados são perdidos.
- As operações lógicas bit-a-bit são aplicadas em tipos char, long e int.

# Conversão de Tipo na Linguagem C

---

- São automáticas as conversões de um tipo “mais estreito” para um “mais largo” sem perda de informações
- Conversões de tipos “mais largos” para tipos “mais estreitos” também são permitidas, mas podem gerar avisos
- Em geral, um operador + ou \* binário com tipos diferentes terá o tipo “menor” promovido para o tipo “maior” resultando no tipo maior

# Conversão de Tipo na Linguagem C

---

- A conversão de tipo pode ser forçada ao se fazer uso do operador unário molde (cast)

<tipo><expressão>

- Exemplo: 

```
int x, y;  
float z;  
.....  
x=10;  
y=3;  
z=(float) x/y;
```

# Conversão de Tipo na Linguagem C

---

- Também é utilizado na passagem de parâmetros em funções

- Exemplo:

```
int x=2;
```

```
...
```

```
sqrt((double) x);
```

# Conversão de Tipo na Linguagem C

---

- Se os argumentos forem declarados em um protótipo de função, a coerção é automática.
- Exemplo:

```
double sqrt(double);  
....  
raiz = sqrt(10);
```

# Exercício I

---

Considere as seguintes variáveis:

- `int i, j, k;`
- `float x, y, z;`

Responda:

- Se `x=12.` e `y=15.`, quanto vale `z = y / x`?
- Se `i=12` e `y=15.`, quanto vale `z = y / i`?
- Se `i=12` e `j=15`, quanto vale `z = j / i`?
- Se `i=12` e `j=15`, quanto vale `z = (float) j / i`?
- Se `i=10` e `j=2`, quanto valem `i, j` e `k` após calculada a expressão..... `k = i++ - --j`?
- Se `i=10` e `j=2`, quanto valem `i` e `j` após calculada a expressão..... `i /= j`?
- Se `x=2.71` e `y=3.2`, quanto valem `x` e `y` após calculada a expressão.... `x += (y++, y / 2)`?

## Exercício II

---

- Faça um programa que leia o nome e as notas de um aluno (nota 1 e nota 2) fornecidas pelo usuário. Usando estas 2 notas, calcule a média simples do aluno, e depois mostre na tela o resultado da média calculada, conforme tela abaixo:

```
>> Calculo da Media <<
```

```
Qual o nome do aluno: Fulano
```

```
Entre com a Nota 1: 8.0
```

```
Entre com a Nota 2: 7.0
```

```
A média das notas do Fulano é 7.50
```

---

## Exercício III

---

- Fazer a leitura de três valores dos coeficientes A, B e C, e depois efetuar o cálculo das raízes de uma equação de segundo grau.

>> Raízes de uma Equação de 2o. Grau <<

Entre com o coeficiente A: 3

Entre com o coeficiente B: 6

Entre com o coeficiente C: 0

As raízes da equação são: -2 e 0.

$$X = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

---

# Referências

---

Ascencio AFG, Campos EAV. Fundamentos de programação de computadores. São Paulo : Pearson Prentice Hall, 2006. 385 p.

Material do Prof. Dr. Fernando Santos Osório do ICMC/USP com teoria e lista de exercícios:

<http://coteia.icmc.usp.br/mostra.php?ident=624>

Outras fontes interessantes

<http://mtm.ufsc.br/~azeredo/cursoC/c.html>

<http://www.cs.cf.ac.uk/Dave/C/>

[http://msdn.microsoft.com/en-us/library/aa315845\(VS.60\).aspx](http://msdn.microsoft.com/en-us/library/aa315845(VS.60).aspx)

[http://www.acm.uiuc.edu/webmonkeys/book/c\\_guide/](http://www.acm.uiuc.edu/webmonkeys/book/c_guide/)

<http://techpubs.sgi.com/library/manuals/0000/007-0701-150/pdf/007-0701-150.pdf>

---

# FIM Aula 4

---