

Fundamentos de Programação

CP41F

Operadores compostos. Operadores relacionais. Precedências.

Aula 8

Prof. Daniel Cavalcanti Jeronymo

Universidade Tecnológica Federal do Paraná (UTFPR)
Engenharia de Computação – 1º Período

- Operadores compostos
 - soma, subtração, multiplicação, divisão, módulo, and, or, xor, SHL e SHR
- Operadores relacionais
 - Igual, diferença, menor, maior, menor-igual, maior-igual
- Precedência e Associatividade
- Comportamento indefinido em atribuições

Operadores compostos

- Combinam atribuição e operação com o operando de lvalue

- variavel OPERADOR= expressao;

Equivale a:

- variavel = variavel OPERADOR expressao;
- +=, -=, *=, /=, %=, >>=, <<=, &=, ^=, |=
- soma, subtração, multiplicação, divisão, módulo, SHR, SHL, and, xor, or

Operadores relacionais

- Testam relações entre expressões
 - $A = 10, B = 20$

Operador	Descrição	Exemplo
==	Verifica se o valor de dois operandos é igual, caso positivo então verdadeiro, caso contrário falso.	$(A == B)$ é falso.
!=	Verifica se o valor de dois operandos é diferente, caso positivo então verdadeiro, caso contrário falso.	$(A != B)$ é verdadeiro.
>	Verifica se o valor do operando à esquerda é maior que o do operando à direita, caso positivo então verdadeiro, caso contrário falso.	$(A > B)$ é falso.
<	Verifica se o valor do operando à esquerda é menor que o do operando à direita, caso positivo então verdadeiro, caso contrário falso.	$(A < B)$ é verdadeiro.
>=	Verifica se o valor do operando à esquerda é maior ou igual que o do operando à direita, caso positivo então verdadeiro, caso contrário falso.	$(A >= B)$ é falso.
<=	Verifica se o valor do operando à esquerda é menor ou igual que o do operando à direita, caso positivo então verdadeiro, caso contrário falso.	$(A <= B)$ é verdadeiro.

Operadores relacionais

- **Atenção!**

O sinal de igual (=) é uma atribuição

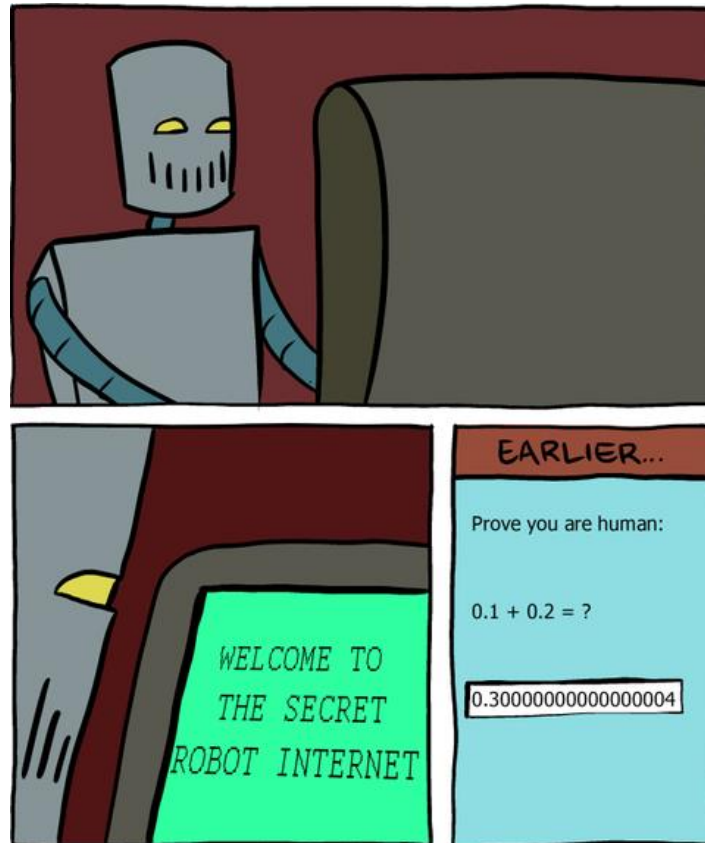
O sinal de igual duplo (==) é um teste de igualdade

- Os operadores relacionais resultam em valores booleanos (1 – Verdadeiro, 0 – Falso)

Operadores relacionais

- **Atenção!**

Comparação de ponto flutuante não deve ser realizada utilizando o operador relacional



Operadores relacionais

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

```
int main()
```

```
{
```

```
    double a = 0.1;
```

```
    double b = 0.2;
```

```
    double c = 0.3;
```

```
    printf("Soma: %f\n", a + b);
```

```
    printf("Comparacao: %s\n", (a + b == c)? "Verdadeiro" : "Falso");
```

```
    printf("a+b: %.18f\n", a + b);
```

```
    printf("c: %.18f\n", c);
```

```
    return 0;
```

```
}
```

Precedência e Associatividade

- Precedência
 - Define a ordem de prioridade em que expressões são avaliadas – $a + b * c \Rightarrow a + (b * c)$
- Associatividade
 - Define como operadores de mesma precedência são agrupados, esquerda-direita (E-D) ou direita-esquerda (D-E)
 - $a = b = c \Rightarrow a = (b = c)$
- Parênteses – forçam a definição de agrupamento das expressões

Precedência e Associatividade

Precedência	Operador	Descrição	Associatividade
1	++	Incremento (sufixo)	E-D
	--	Decremento (sufixo)	
2	++	Incremento (prefixo)	D-E
	--	Decremento (prefixo)	
	+	Positivo (unário)	
	-	Negativo (unário)	
	!	Not lógico	
	~	Not binário	
	(type)	Type cast	
	*	Dereferência	
	&	Endereço	
	sizeof	Size-of	
3	*	Multiplicação	E-D
	/	Divisão	
	%	Resto (módulo)	
4	+	Adição	E-D
	-	Subtração	
5	<<	Deslocamento à esquerda (SHL)	E-D
	>>	Deslocamento à direita (SHR)	
6	<	Menor	E-D
	<=	Menor igual	
	>	Maior	
	>=	Maior igual	

Precedência	Operador	Descrição	Associatividade
7	==	Igualdade	E-D
	!=	Desigualdade	
8	&	And binário	E-D
9	^	Xor binário	E-D
10		Or binário	E-D
11	&&	And lógico	E-D
12		Or lógico	E-D
13	?:	Condicional	D-E
14	=	Atribuição	D-E
	+=	Soma	
	-=	Subtração	
	*=	Multiplicação	
	/=	Divisão	
	%=	Resto	
	<<=	SHL	
	>>=	SHR	
	&=	And binário	
	^=	Xor binário	
	=	Or binário	

Ordenados por ordem decrescente de prioridade

Precedência e Associatividade

- Precedência
 - Define a ordem de prioridade em que expressões são avaliadas – $a + b * c \Rightarrow a + (b * c)$
- Associatividade
 - Define como operadores de mesma precedência são agrupados, esquerda-direita (E-D) ou direita-esquerda (D-E)
 - $a = b = c \Rightarrow a = (b = c)$
- Parênteses – forçam a definição de agrupamento das expressões

Comportamento indefinido

- Pontos de sequência (sequence points) – pontos onde todas as avaliações de expressões foram completas e não há efeitos penderentes
 - Avaliação de curto-circuito nos operadores lógicos AND e OR, declarações, etc
- Entre o ponto de sequência anterior e próximo, um objeto escalar terá seu valor modificado no máximo uma vez
- `a = a++`; problema!

Comportamento indefinido

- Avaliação de curto-circuito
 - Permite encerrar uma operação lógica sem avaliar todos os elementos
- Exemplos:
 - $A == A \ || \ \text{Expressão} - \text{Verdadeiro}$
 - $A != \&\& \ \text{Expressão} - \text{Falso}$