

Fondamenti di Informatica e Laboratorio T-AB
Ingegneria dell'Automazione
a.a. 2008/2009

Lab 03

Precedenza e associatività negli
operatori. Assegnamento,
incremento, overloading e casting

Esercizio 5

Conversione, Coercizione, Overloading

Si assegnino valori iniziali alle variabili nel modo seguente:

```
int A <- 10   unsigned short int B <-104   char C <- 77   double D<-10.0
```

Si eseguano le seguenti operazioni, ripristinando i valori iniziali delle variabili dopo ogni istruzione:

1. $C = C \div D * D$
2. $C = C \div A * A$
3. $C = C * A \div A$
4. $C = C * A$ seguita da $C = C \div A$ (2 istruzioni separate)
5. $D = D \div 3 * 3$
6. $A = A \div 3 * 3$
7. $A = (B \div (C + 3)) \div D$
8. $C =$ resto della divisione di D per 2

Si controlli l'esecuzione del programma con in debugger, per visualizzare il contenuto delle variabili.

Che valori visualizza il debugger al termine dell'esecuzione di ciascuna istruzione?

Sono quelli che ci si aspetta?

In quali casi non lo sono? Perché accade ciò?

Esercizio 6

Overflow, Underflow e Divide by 0

Siano date le variabili viste prima, più due nuovi float: E ed F.

Si assegnino valori iniziali alle variabili nel modo seguente:

A <- -10 B <- -10⁴ C <- -77 D <- 10.0 float E, F <- 10⁻¹⁰

Si eseguano le seguenti operazioni, ripristinando i valori iniziali delle variabili dopo ogni istruzione:

1. C = C * 2
2. B = C * B
3. A = B * B * B
4. F = D * D seguito da F = F ÷ D ÷ D
5. E = E * E seguito da D = D ÷ E ÷ E
6. F = A + D seguito da F = F=A
7. D = A + D seguito da D = D=A
8. D = A + D seguito da A = A=D seguito da B = B=A
9. E = A + E seguito da A = A=E seguito da B = B=A
10. A = A=(A + E) seguito da B = B=A

Si controlli l'esecuzione del programma con in debugger, per visualizzare il contenuto delle variabili.

Si descriva e motivi il risultato di ciascuna istruzione.

Esercizio 7

Casting

Si considerino le espressioni dei punti 5 e 6 che davano risultati non corrispondenti all'aritmetica dei numeri interi e reali.

Si utilizzi il meccanismo del casting esplicito per far sì che non si verifichino problemi di troncamento, tipi degli operandi, overflow, underflow, e divisione per zero ove possibile (non è detto che si riesca ad evitare il problema) e non modificando il tipo delle variabili:

- A è di tipo int.
- B è di tipo unsigned short int;
- C è di tipo char;
- D è di tipo double;
- E, F sono di tipo float.

Operatori: priorità ed associatività

Priorità	Operatore	Simbolo	Associatività
1 (max)	chiamate a funzione selezioni	() [] -> .	a sinistra
2	operatori unari: op. negazione op. aritmetici unari op. incr. / decr. op. indir. e deref. op. sizeof	! ~ + - ++ -- & * sizeof	a destra
3	op. moltiplicativi	* / %	a sinistra
4	op. additivi	+ -	a sinistra

Operatori: priorità ed associatività

Priorità	Operatore	Simbolo	Associatività
5	op. di shift	>> <<	a sinistra
6	op. relazionali	< <= > >=	a sinistra
7	op. uguaglianza	== !=	a sinistra
8	op. di AND bit a bit	&	a sinistra
9	op. di XOR bit a bit	^	a sinistra
10	op. di OR bit a bit		a sinistra
11	op. di AND logico	&&	a sinistra
12	op. di OR logico		a sinistra
13	op. condizionale	? . . . :	a destra
14	op. assegnamento e sue varianti	= += -= *= /= %= &= ^= = <<= >>=	a destra
15 (min)	op. concatenazione	,	a sinistra

Esercizio 8

Si riconsideri la seguente tabella degli operatori in C: Precedenza
Operatori Associatività

Siano date le seguenti definizioni di variabile:

```
typedef unsigned short int _us;  
int A=1; _us B=20; char C='k'; float E=11.5;
```

Come vengono valutate le seguenti espressioni?

1. Expr1: (int) E % B / 2 != C (A = B * C || B % 2)

Si verifichi con il debugger la correttezza delle risposte date.