

Espressioni

Forma sintattica (informale)

Una espressione (*expr*) può essere:

- una costante
- una variabile
- un'espressione composta della forma
expr operatore_binario expr

dove *operatore_binario* è un operatore aritmetico o di confronto o logico

- un'espressione tra parentesi
(*expr*)
- uno stmt di assegnamento
- una chiamata di funzione
- ...

Valutazione di un'espressione

Determinare il valore di un'espressione applicando gli operatori ai loro operandi, in un ordine opportuno

Semantica operatori: (nella maggior parte dei casi) quello intuitivo; es.: + per somma, && per and logico, ...

Ordine valutazione: agli operatori si applicano le usuali regole di precedenza e associatività

Precedenza tra operatori (circa 60 operatori primitivi in C++)

Operatori raggruppati in classi di precedenza (18 classi): operatori con precedenza più alta valutati per primi.

Tabella precedenze operatori (parziale):

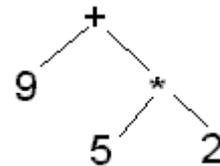
<i>Prec. più alta</i>	!	++	--
	*	/	%
	+	-	
	<	>	<= >=
	&&		
<i>Prec. più bassa</i>	=		

Es.: 9 + 5 * 2

poiche' il * ha precedenza sul + prima valuta 5 * 2 e poi 9 + 10

corrisponde all'albero sintattico:

prima valuta i sottoalberi più in profondità, da sinistra a destra

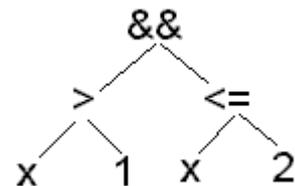


Es.: x > 1 && x <= 2

prima valuta x > 1 e x <= 2 e poi il &&

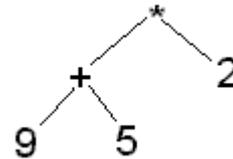
corrisponde all'albero sintattico:

n.b.: non richiesto uso di parentesi attorno alle sottoespress.



L'uso esplicito di parentesi permette di controllare l'ordine di valutazione: espressione tra parentesi assume massima priorità

Es.: $(9 + 5) * 2$ corrisponde a



Associatività: tra operatori con la stessa precedenza si procede solitamente da sinistra a destra (operatori associativi a sinistra)

Es.: $9 + 2 - 5$ come se fosse $(9 + 2) - 5$

$6 / 2 * 0$ come se fosse $(6 / 2) * 0$

Gli operatori unari e quelli di assegnamento invece sono associativi a destra.

Es.: $x = y = z = 1$ come se fosse $x = (y = (z = 1))$

Tipo di un'espressione

Il tipo di un'espressione è il tipo del suo risultato.

Per operatori "sovraccarichi" (ad es. $+$, $*$, $-$, ...) il tipo del risultato dipende dal tipo degli operandi

Es.:

Da completare