INFORMATICA

MASTER COPY

Tel. 388/9837745

prof. Lamfranco Lopriore

Esame scritto - orale (12 crediti)

LINGUAGGIO C++

Il linguaggio C++ è un'estensione del linguaggio C; lo sviluppo del linguaggio c++ all'inizio degli anni Ottanta è dovuto a Bjarne Stroupstroup dei laboratori Bell ed è un linguaggio di alto livello.

STRUTTURE DEL LINGUAGGIO:

GLI IDENTIFICATORI: sono i nomi utilizzati per rappresentare variabili, costanti, tipi e funzioni del programma. Un identificatore è formato da una sequenza di una o più lettere, cifre o caratteri e deve sempre iniziare con una lettera o con un carattere di sottolineatura; gli identif. possono contenere un qualunque numero di caratteri ma solo i primi 31 sono significativi per il compilatore. Il linguaggio C++ distingue le lettere maiuscole dalle minuscole.

VARIABILI: esse sono altro che contenitori di informazione in altre parole la variabile è l'astrazione del concetto di cella di memoria.

TIPI STANDARD DEL C++:

- CHAR: per rappresentare i tipi di testo; può essere formato da serie di caratteri (b, F, !, ?, 6) e stringhe ("ciao mondo") il tipo char occupa 8 bit in altri termini 1 byte a carattere
- INT: per rappresentare i valori interi (1, 2, 43, -23 ecc) nei sistemi a 32 bit il tipo int occupa 32 bit ovvero 4 byte e quindi contiene valori compresi tra -2.147.483.648 e 2.147.483.648.
- FLOAT: per rappresentare i valori in virgola mobile, utilizzati per rappresentare i numeri decimali e sono rappresentati da una parte intera e una frazionale. I numeri in virgola mobile richiedono una maggiore precisione rispetto agli int e sono rappresentati da 32 bit e il loro intervallo varia tra i $3,4 \times 10^{34}$ e $3,4 \times 10^{38}$ (con 7 cifre significative)
- DOUBLE: per rappresentare valori in virgola mobile più estesa che occupa 64 bit (o 8 byte) $[1,7 \times 10^{-308}$ e $1,7 \times 10^{308}]$ (15 cifre sign.) poi abbiamo long double che occupa invece 80 bit (10 byte)
- VOID: rappresenta valori che occupano 0 byte e non hanno alcun valore
- ENUM: definiti dall'utente...

MASTER COPY

Tel. 388/9837745

es. `int x = 0;`
- `int` → tipo
- `x` → Variabile "contenitore" valore sinistro
- `0` → "contenuto" valore destro, valore assegnato alla variabile

SHORT, LONG, UNSIGNED: sono qualificatori che si aggiungono al tipo per aumentare il significato. Abbiamo detto che i tipi formati da un byte ecc; un byte a sua volta è formato da 8 bit. Ed il bit (binary digit ovvero cifra binaria) può contenere i valori 0 e 1. Per esempio 10100010
Si definisce bit più significativo quello all'estrema sinistra
Nella notazione normale, senza l'uso di qualificatori, il bit più significativo viene utilizzato per il segno. In particolare se il primo bit vale uno rappresenta il segno meno mentre lo 0 il segno +. Quindi invece di 8 bit per rappresentare un valore ne verranno usati solo 7. Utilizzando il qualificatore unsigned tutti i bit che compongono il byte vengono utilizzati per contenere il valore che in tal caso sarà solo positivo.
Il qualificatore short equivale fondamentalmente al tipo stesso; dire int o dire short int è equivalente. mentre il qualificatore long è invece usato per incrementare il range di valori che un int può contenere.

long double = 80 bit (10 byte)
int = long int = 32 bit
nei sistemi a 32 bit

CONST: In molti casi è utile assegnare a degli identificatori dei valori che restino costanti durante tutto il programma. In C++ è possibile con il modificatore const; es `const int i = 3;`

ENUM: Quando si definisce una variabile di tipo enumerativo, ad esso viene associato un insieme di costanti intere chiamato insieme dell'enumerazioni
ad esempio `enum Giorno {LUN, MAR, MER, GIO, VEN, SAB, DOM};`
- Crea un nuovo tipo enum Giorno
- Nota bene: la prima lettera deve essere maiuscola
- LUN, MAR, ... sono costanti intere che se non inizializzate valgono 0 la prima, 1 la seconda ecc.

```
Giorno oggi = LUN
cout << LUN;
```
- oggi → variabile enumerativa
- In questo caso stampa 0 (conversione implicita)

PROPRIETÀ DEGLI OPERATORI: In un'espressione con più operatori esistono delle proprietà degli operatori per determinare quali operatori vengono prima d'altri:

① ARITÀ = numero di operanti
- a = (operatore unario, 1 operando)
a - b = (operatore con due operanti)

② POSIZIONE
prefisso "-a"
infisso a-b
postfisso a++

③ PRECEDENZA O PRIORITÀ
* (per) ha priorità sul +
le parentesi in C++ vengono considerate operatori

④ ASSOCIATIVITÀ
In una espressione in presenza di due operatori con la stessa priorità si valuta da sinistra verso destra.

es j = 3*1-3/3 = scritto così prima è scritto quale usare sempre le parentesi!

3/2*5

j = 3*(1-3)/3 =

MASTER COPY
Tel. 388/9837745

OPERATORI RELAZIONARI E LOGICI sono &&, ||, !

AND && (congiunzione logica): prende in input due operandi e produce in output True o false (V o F) seguendo questa regola: Se entrambi gli operatori sono true allora l'output sarà true, in tutti gli altri casi l'output sarà false

OR || (disgiunzione logica): prende in input due operandi e produce in output un booleano; Se almeno uno degli operandi è uguale a true, l'output sarà true; altrimenti se nessuno è uguale a true l'output sarà false.

NOT ! (negazione logica): prende in input un solo operando e produce in output un booleano; Se l'operando di input è True allora l'output sarà false e viceversa in altre parole l'operatore NOT prende un input e me restituisce il contrario.

1	2	&&	||
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	F

1	!
V	F
F	V

In C++ il valore "0" equivale a FALSE e qualunque valore diverso da zero equivale a TRUE

|| e && sono gli unici operatori in cui la valutazione avviene prima nel valore sinistro e poi nel destro; se il valore sinistro permette di risolvere lo statement il valore destro viene ignorato "cortocircuito"

Un operatore di relazione è un operatore che verifica una condizione

MAGGIORE: si indica con il simbolo >
MINORE: si indica con il simbolo <
UGUALE: si indica con il simbolo ==
MAGGIORE O UGUALE: si indica con il simbolo >=
MINORE O UGUALE: " " " " " <=
DIVERSO: " " " " " !=

} restituiscono un valore booleano ovvero true o false

OPERATORI ARITMETICI: Il linguaggio C++ è dotato di tutti i comuni operatori aritmetici di somma +, sottrazione -, divisione /, moltipl* e modulo %. L'operatore modulo % restituisce il resto di una divisione es. int a=10, b=5
a%b = 0

LIVELLI DI PRECEDENZA DEGLI OPERATORI: indicano l'ordine con cui gli operatori vengono valutati dal compilatore.

OPERATORE	NOME
!	NOT
*	MOLTIPLICAZIONE
/	DIVISIONE
%	MODULO
+	ADDIZIONE
-	SOTTRAZIONE
<	MINORE
<=	MINORE UGUALE
>	MAGGIORE
>=	MAGGIORE UGUALE
==	UGUALE
!=	DIVERSO
&&	AND
||	OR
=	ASSEGNAMENTO

Le parentesi risolvono il problema delle priorità.

la tabella è suddivisa in gruppi. tutti gli operatori appartenenti ad uno stesso gruppo hanno lo stesso livello di precedenza mentre gli operatori appartenenti ad un gruppo hanno maggior precedenza di tutti gli altri operatori che compaiono nei gruppi sottostanti.

IF ed ELSE: è un'istruzione condizionale. La sintassi è:

```
if (condizione) {
    istruzioni da svolgere se la condizione è vera }
else { istruzioni da svolgere se la condizione è falsa }
```

In caso di if in cascata si dice che un'istruzione else è sempre riferita all'ultima istruzione if (ovvero quello più vicino). L'if è un'istruzione di controllo che, in base alla condizione, può modificare l'ordine sequenziale dell'algoritmo. Una condizione è sempre un numero intero, se è falso è uguale a zero se è vero un qualunque numero.

?: l'istruzione condizionale ? fornisce un modo rapido per scrivere una condizione di test; le azioni associate all'istruzione condizionale vengono eseguite in base al valore dell'espressione (true o false). Quindi l'operatore ? può essere utilizzato per sostituire un'istruzione if-else. La sintassi di ? è:

espressione test ? azione true : azione false.

L'operatore ? viene anche denominato operatore Ternario.

SWITCH: le istruzioni switch permettono di evitare molte sequenze di if. E' utile quando in un programma si deve dare all'utente la possibilità di scegliere tra più opzioni. La sintassi è:

```
switch (espressione intera) {
case (valore costante 1):
    sequenze di istruzioni
case (valore costante 2):
    seq. di istruz.
case (val. cost 3):
    ...
default:
    sequenza di istr. }
```

MASTER COPY
Tel. 388/9837745

Il costrutto esegue le seguenti operazioni:

- valuta il valore dell'espres. intera passata come parametro all'istruzione switch soddisfa la condizione dell'istruz.
- rimanda lo svolgimento del programma al blocco in cui il parametro
- se nessun blocco corrisponde ad un valore uguale a quello dell'istruz. switch allora viene eseguito il blocco default, se presente.