Corso sul linguaggio C++

Modulo LC

1.2.2 – Operare con i dati

Prerequisiti

- Concetto matematico di funzione
- Concetto matematico di insieme
- Concetto di operazione e di espressione

Introduzione

In questa Unità studiamo le operazioni principali, la loro classificazione e il loro effetto.

Informazioni generali

Combinando costanti e variabili, appartenenti ad un determinato tipo di dato T **(int, float**,…), mediante opportuni **operatori** (*ammessi per il tipo T*) possiamo costruire **espressioni** di tipo T.

Vediamo in ordine questi concetti.

Le variabili
- nome delle variabili

Per i nomi delle variabili si possono utilizzare:

- lettere maiuscole (A..Z)
- lettere minuscole (a..z)
- sottolineatura (_)
- cifre (0..9)

È buona abitudine:

- dare alle variabili nomi significativi
- scrivere le variabili in caratteri minuscoli

Le variabili
- dichiarazione delle variabili

Esempi di dichiarazione di variabili:

```
int conteggio, valore_iniziale;
float somma1;
double asse_maggiore,
       asse_minore;
```

N.B. – Il linguaggio C++ è *case-sensitive* ossia distingue i caratteri maiuscoli da quelli minuscoli. Quindi le variabili Asse, asse, ASSE, sono considerate diverse.

Le variabili
- variabili e compilatore

Una delle maggiori caratteristiche delle tecnologie informatiche è la possibilità di lavorare con **linguaggi simbolici** ad **alto livello**.

Ciò significa che il programmatore può comodamente lavorare con **variabili simboliche**, invece che trattare con '0' ed '1'.

Sarà poi il **compilatore** a trasformare i nomi delle variabili nei rispettivi indirizzi di memoria **RAM** (dopo aver controllato l'assenza di errori sintattici nel programma)

Le variabili
- tabella dei simboli

Il compilatore nella sua **tabella dei simboli** mantiene traccia della corrispondenza tra i nomi delle variabili usate dal programmatore con i loro rispettivi indirizzi di memoria.

Tabella dei simboli del compilatore

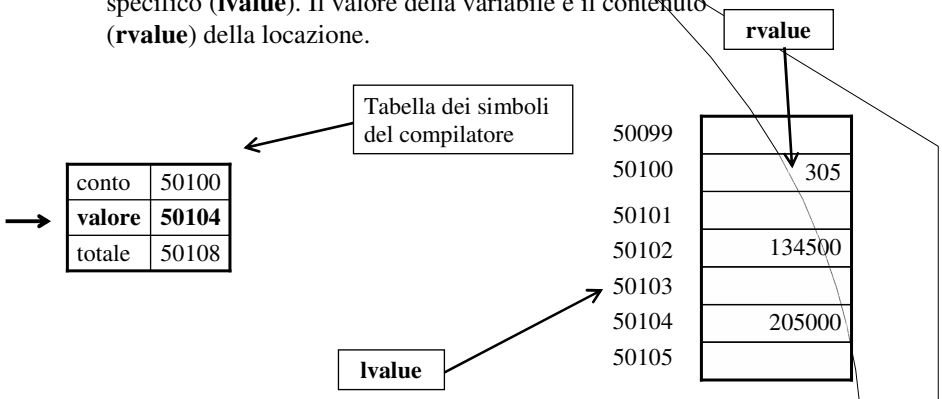
conto	50100
valore	50102
totale	50104

50099	
50100	305
50101	
50102	134500
50103	
50104	205000
50105	

In questo esempio supponiamo che ogni locazione di memoria sia di un byte e che un numero intero occupi 2 byte in memoria

Le variabili
- lvalue e rvalue

Ogni variabile è allocata im memoria ad un indirizzo specifico (**lvalue**). Il valore della variabile è il contenuto (**rvalue**) della locazione.



Le variabili
- usi di sizeof()

L'occupazione in memoria di un qualunque tipo di dato è rilevabile mediante la funzione **sizeof**(), come mostrato nell'esempio

```
// size.cpp
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{   cout<<"Type\t    N^ byte\n";
    cout<<"-------------------\n";
    cout<<"char\t\t"<<sizeof (char)<<endl;
    cout<<"int\t\t"<<sizeof (int)<<endl;
    cout<<"float\t\t"<<sizeof (float)<<endl;
    cout<<"double\t\t"<<sizeof (double)<<endl;
    system("Pause");
    return EXIT_SUCCESS;
}
```

Operatori

Sulle variabili di ogni tipo di dato (**char, int, float, double**, ecc) possiamo effettuare delle elaborazioni mediante gli **operatori**, scrivendo delle **espressioni**.

Ogni operatore è caratterizzato da:

- un **simbolo** o un nome (ad es. +, -, *, /)
- l'**operazione** svolta (addizione, sottrazione, ..)
- determinate **regole di precedenza** tra operatori che servono a stabilire le priorità nel calcolo delle espressioni (che vedremo fra breve)

Operatori
- classificazione

Gli operatori che studiamo sono i seguenti:

- Assegnamento
- Operatori aritmetici
  - Operatori interi
  - Operatori reali
- Operatori relazionali
- Operatori logici
- Operatori sul bit

Assegnamento

L'operatore di **assegnamento**, che già conosciamo, è di fondamentale importanza. Esso consiste nell'attribuire ad una variabile un dato valore.

L'assegnamento si indica con il simbolo "=" e la sua sintassi è:

ident = *espressione*;

dove:

ident è il nome che abbiamo scelto per la variabile;

espressione è una combinazione di operatori e variabili (tra breve ne daremo una definizione più precisa)

Assegnamento

- Esempi:
  - c = 'A'; /* assegna a *c* la costante 'A' */
  - cognome = "Rossi"; /* alla variabile cognome assegna la stringa costante "Rossi" */
  - conteggio = 0; /* assegna il valore costante 0 */
  - raggio = 0.5256; /* assegna il valore costante 5.256 */
  - a = b = c; /* assegnazione multipla */
  - Raggio =5.256E-01; /* assegnazione di un valore reale */

Operatori aritmetici interi

Operatori **binari** (agiscono su due variabili) **interi**:

- Addizione (+)
- Sottrazione (-)
- Moltiplicazione (*)
- Divisione (/)
- Resto (%)

Operatori **unari** (agiscono su una sola variabile) **interi:**

- Autoincremento (++)
- Autodecremento (--)

Autoincremento e autodecremento

Ad esempio, date le variabili:
int a=3,b=2,c;
Si ha:

Postincremento: prima assegna il valore e poi incrementa a di 1

Preincremento: prima incrementa di 1 il valore di a e poi lo assegna a c

Istruzione	a	b	c
c=a+b;	3	2	5
c=a-b;	3	2	1
c=a*b;	3	2	6
c=a/b;	3	2	1
c=a%b;	3	2	1
c=a++;	3	2	3
c=++a;	4	2	4
a++;	5	2	
b--;	2	1	

a++; equivale ad **a=a+1;**
a- -; equivale ad **a=a-1;**

Operatori aritmetici reali

Questi operatori agiscono su variabili **reali**:

- Addizione (+)
- Sottrazione (-)
- Moltiplicazione (*)
- Divisione (/)
- Funzioni matematiche (presenti nella libreria **math.h**).
  - Ad es. **abs**(), **max**(), **min**(), **sin**(), **cos**(), **exp**(), **log**(), **pow**(), **rand**(), **sqrt**(), **tan**().

Operatori aritmetici reali

ATTIVITA': Scrivere un programma che, note le misure a e b dei cateti di un triangolo rettangolo, calcoli e stampi il perimetro.

```
// Pitgora.cpp
#include <iostream>
#include <cmath>
using namespace std;
int main()
{ float a,b,p,c;
  cout<<"Primo cateto: ";
  cin>>a;
  cout<<"Secondo cateto: ";
  cin>>b;
  c = sqrt (pow (a, 2)+ pow (b, 2));
  p=a+b+c;
  cout<<"Il valore del perimetro e' "<<p<<endl;
  system("Pause");
  return 0;
}
```

Operatori relazionali

Questi operatori agiscono in genere su variabili **reali** o **intere**:

- Maggiore (>)
- Maggiore o uguale (>=)
- Minore (<)
- Minore o uguale (<=)
- Uguale (= =)
- Diverso (!=)

Operatori logici

Gli operatori logici agiscono su variabili logiche che come abbiamo detto, sono variabili che possono assumere soltanto uno tra due possibili valori (**true** o **false).**

- Oltre alle variabili dichiarate esplicitamente di tipo **bool**, possiamo considerare **variabili logiche** del tipo seguente:

  a == 5

  dove il simbolo "= =" indica chiedersi se l'uguaglianza scritta è vera.

  Esempio:

```
bool esito;          // dichiara la variabile booleana esito
int a = 5;
esito = (a = = 5);   // esito assume valore true
```

Operatori logici

Gli operatori logici sono:

- **AND** (&&) detto **prodotto logico**
- **OR** (| |) detto **somma logica**
- **NOT** (!) detto **negazione**

In C++ i valori logici sono associati ai valori **0** e **1** nel seguente modo:

VERO	**1**
FALSO	**0**

Tabella di verità

Il modo di operare degli operatori logici viene descritto mediante la loro **tabella di verità**.

ATTIVITA': Disegnare la tabella di verità dell'**AND**.

Tabella di verità dell'operatore **AND**

&&	**0**	**1**
0	**0**	**0**
1	**0**	**1**

ATTIVITA': Disegnare la tabella di verità dell'**OR**.

Tabella di verità dell'operatore **OR**

||	**0**	**1**
0	**0**	**1**
1	**1**	**1**

ATTIVITA': Disegnare la tabella di verità del **NOT**.

Tabella di verità dell'operatore **NOT**

!	
0	**1**
1	**0**

Tabella di verità

```
// OperatoriVariabiliLogiche.cpp
# include <iostream>
#include <cstdlib>
using namespace std;
int main(int argc, char *argv[])
{  int a=5, b=4;
   bool a_pari, b_pari;
   a_pari = (a%2==0);
   b_pari = (b%2==0);
   cout<<a<<" e' pari? "<<a_pari<<endl;
   cout<<b<<" e' pari? "<<b_pari<<endl;
   system("Pause");
   return EXIT_SUCCESS;
}
```

Esempio di uso di variabili **bool**

Stampa 0 perché a non è pari

Stampa 1perché b è pari

Operatori sul bit

Esistono in particolare operatori in grado di lavorare bit a bit su un dato

&	**And**
|	**Or**
^	**Or esclusivo**
~	**Not**
<<	**Shift a sinistra**
>>	**Shift a destra**

Operatori abbreviati

Per molti operatori esiste una forma abbreviata come mostrato in tabella

Forma abbreviata	**La forma…**	**…sta per**
+= -= *= /= %=	x+=a;	x=x+a;
&= |= ^=	x&=a;	x=x&a;
<<= >>=	x<<=y;	x=x<<y

Espressioni

Variabili e **costanti** di un certo tipo (**int**, ecc) possono essere legati tra loro mediante gli operatori del medesimo tipo per creare **espressioni.**

Possiamo scrivere espressioni per ogni tipo di dato, rispettando la sintassi degli operatori previsti.

Una **espressione** può essere:

- una **costante**
- una **variabile**
- una **relazione** fra espressioni

Esempi di espressioni

Espressione	Tipo espressione
5+3	Aritmetica Intera
b<c || t!=r	Logica (vera se b<c o t diverso da r)
!y	Logica (vera se y = 0)
x= =y && a>b	Logica (vera se x=y e a>b)
a+b%2	Aritmetica intera
5+2+(i<10)	Aritmetica intera (vale 8 se i<10, 7 altrimenti)
c	Variabile intera (se c è una variabile intera)
'a'	Costante di tipo carattere
"Rossi"	Costante di tipo stringa
rapporto / 2.0	Variabile float (se rapporto è di tipo float)
0	Costante intera

Precedenza tra gli operatori

Precedenza (*)	Simbolo	Operazione
1	!	Negazione
2	* / %	Moltiplicazione, divisione, resto
3	+ -	Addizione, sottrazione
4	> >= < <=	Maggiore, minore, ecc. ecc
5	== !=	Uguale, diverso
6	&&	And
7	||	Or
8	?:	Operatore ternario (**)
9	=	Assegnazione

(*) L'ordine di priorità può essere alterato mediante l'uso di parentesi

(**) L'operatore ternario '?:' verrà illustrato nella Lezione "Strutture di controllo".

Modificatori dei tipi di dati

Esistono delle parole chiave che modificano il tipo di dato.

- **unsigned**

  Per gli interi consente capacità doppia perché il segno non viene usato. Il default è **signed**

- **short**

  Per gli interi usa una quantità di memoria pari alla metà (maggiore velocità nei calcoli)

Argomenti

- Le variabili
  - nome delle variabili
  - dichiarazione delle variabili
  - variabili e compilatore
  - tabella dei simboli
  - lvalue e rvalue
  - uso di **sizeof**()
- Operatori
  - classificazione
- Assegnamento
- Operatori aritmetici interi
- Autoincremento e autodecremento
- Operatori aritmetici reali
- Operatori relazionali
- Operatori logici
- Tabella di verità
- Operatori sul bit
- Operatori abbreviati
- Espressioni
- Esempi di espressioni
- Precedenza tra gli operatori
- Modificatori dei tipi di dati

Altre fonti di informazione

- J. Purdum, C – ed. Jackson
- Romagnoli-Ventura, C/C++ - Ed. Petrini
- A.Bellini-A.Guidi, Conoscere il C – ed. McGraw Hill
- A. Lorenzi et alii – Il linguaggio C++ - Ed. ATLAS