

Corso di Fondamenti di Informatica Istruzioni Semplici

Prof. Valeria Vittorini



DIS - Dipartimento di Informatica e Sistemistica - Università di Napoli

Riepilogo di alcuni concetti fondamentali

- Programma
 - Insieme di istruzioni che, eseguite nell'ordine specificato, trasformano, secondo la funzione F , i dati d'ingresso X nei dati d'uscita Y
- Dati
 - Costituiscono l'oggetto dell'elaborazione
 - Possono appartenere ai tipi predefiniti del linguaggio o a tipi definiti dall'utente
- Sezione dichiarativa
 - Definisce i tipi, le variabili e le costanti del programma
- Variabili
 - Contengono informazioni che vanno soggette a modifiche durante l'esecuzione del programma
- Costanti
 - Contengono informazioni che rimangono inalterate durante l'esecuzione del programma

Tassonomia delle istruzioni

- Le istruzioni esprimono fundamentalmente due classi di azioni elaborative:
 - alterazione di valori di variabili e colloquio con unità periferiche
 - Controllo di sequenza
- Istruzioni semplici
 - Rappresentano azioni elementari del programma
- Istruzioni strutturate
 - Sono composte da azioni organizzate in strutture
 - Le azioni componenti possono essere istruzioni semplici o istruzioni a loro volta strutturate

Istruzioni semplici

- Istruzioni di ingresso
- Istruzioni di uscita
- Istruzioni di calcolo ed assegnazione
- Istruzioni di salto semplice o condizionato
- Istruzioni di chiamata a sottoprogramma

Istruzioni d'ingresso

- “Leggono” da una periferica e “scrivono” su una variabile
 - Esempi:
 - Lettura di un valore da terminale
 - Lettura di un file da disco
 - Lettura di un dato dalla rete
- Specificano (tipicamente):
 - L'unità d'ingresso
 - Il formato dei dati
 - La/e variabile/i su cui operano

Istruzioni di ingresso

- I dati sono registrati o digitati secondo un formato
esempio:
 - 3.4+i8
 - (-3.4,8)
- Sono spesso utilizzate alcune convenzioni implicite (default):
 - sull'unità di ingresso
 - sul formato dei dati
- In questo caso una istruzione di ingresso ha la forma generale:

read *lista_variabili*

Istruzioni di ingresso

cin >> variabile;

- è una istruzione di ingresso in C++ dove l'unità di default è la tastiera e il formato dei dati coincide con il formalismo previsto per il tipo corrispondente
- l'operatore di input >> consente di eseguire operazioni di input su tipi predefiniti
- La lettura di una lista di variabili si ottiene con la forma: `cin >> var_1 >> var_2 >> ...>> var_n;`

Istruzioni d'uscita

- “Leggono” da una variabile e “scrivono” su una periferica
 - Esempi:
 - Scrittura di un valore da terminale
 - Scrittura di un file su disco
 - Invio di un dato in rete
- Specificano (tipicamente):
 - L'unità d'uscita
 - Il formato dei dati
 - La/e variabile/i su cui operano

Istruzioni di uscita

- I dati sono "trascritti" secondo un formato
esempio:
-3.4+i8
(-3.4,8)
- Sono spesso utilizzate alcune convenzioni implicite (default):
 - sull'unità di uscita
 - sul formato dei valori
- In questo caso una istruzione di uscita ha la forma generale:

write *lista_valori*

Istruzioni di uscita

cout << espressione

- è una istruzione di uscita in C++ dove l'unità di default è il video e il formato dei valori coincide con il formalismo previsto per il tipo corrispondente
- l'operatore di output << consente di eseguire operazioni di output su tipi predefiniti
- La stampa (a video) di una lista di valori si ottiene con la forma: `cout <<exp_1 << exp_2 << ...<< exp_n;`

Esempi di istruzioni di I/O

```
cout << endl << "Immettere il valore della variabile x: ";  
cin >>x;
```

```
f.read((unsigned char*)&lib, sizeof(Libro));  
f.write((unsigned char*)&lib, sizeof(Libro));
```

```
printf("Scelta ---> ");  
scanf("%c",&ch);
```

```
read(ssock,filename,sizeof(filename));  
write(ssock,&msg,sizeof(msg));
```

Esempi (*cin* e *cout*)

- int i; cin >> i;
- char s[10]; cin >> s;
- cout << "abcd";
- cout << '\n';
- cout << 'a';
- int a=0; cout << "il valore è: " << a;

Istruzioni di calcolo e di assegnazione

- Portano il valore corrente di una variabile al valore di una costante o al risultato di un'espressione
- Il valore della variabile è anche detto "left value", quello dell'espressione "right value"
- Le regole di composizione dipendono da numerosi fattori, tra cui:
 - Gli operandi presenti nell'espressione ed il loro tipo
 - Gli operatori previsti per ciascun tipo
 - La precedenza degli operatori
 - Le conversioni – implicite ed esplicite – tra tipi

Istruzioni di calcolo ed assegnazione

- memorizzano il risultato in un registro di memoria associato ad una variabile
- In generale:

variabile:=espressione

Left value

right value

- In altre parole: una **variabile** è un **simbolo** associato a un registro di memoria o *indirizzo fisico* (**L-VALUE**) che contiene un valore (**R-VALUE**)

Calcolo ed assegnazione in C++

- **variabile=espressione**
- **operatori di incremento e decremento:**
 - `i++` sta per `i=i+1` notazione postfissa
 - `++i` sta per `i=i+1` notazione prefissa
 - `i--` sta per `i=i-1` notazione postfissa
 - `--i` sta per `i=i-1` notazione prefissa
- **operatori di assegnazione composta:**
 - `a<op>=b` sta per `a=a<op>b`
 - esempi: `a+=b` sta per `a=a+b`
 - `a*=b` sta per `a=a*b`
 - ...

The image shows a screenshot of a C++ development environment. At the top, a console window displays the output of a program: "1", "2", and "Prendere un tasto per continuare . . .". Below it, the main code editor shows the following C++ code:

```

1 #include <iostream>
2 #include <stdlib.h>
3
4 using namespace std;
5
6 int main(int argc, char *argv[])
7 {
8     int a=3;
9     int b=1,c=1;
10    cout << "\n" << b++ << "\t" << ++c;
11    cout << "\n";
12    system("PAUSE");
13    return 0;
14 }

```

The IDE interface includes a menu bar (File, Modifica, Cerca, Visualizza, Progetto, Esegui, Debug, Strumenti, CVS, Finestra, Help), a toolbar, and a status bar at the bottom showing "11: 16" and "14 Righe nel file".

Esempi di istruzioni di assegnazione

- Nell'ipotesi che tutte le entità in gioco siano state correttamente definite:
- $A=B$;
- $A=3$;
- $A=4+B$;
- $A=B+C$;
- $A=A+B$;
- $A=F(B)$;
- $A=B++$

Type checking

- Deve esserci "compatibilità" tra la variabile e il valore da assegnarle
- **Type checking** significa controllare che ciascuna operazione eseguita da un programma riceva il numero appropriato di argomenti, e del tipo atteso.
- Diverse soluzioni per il type checking:
 - a tempo di esecuzione o a "run time" (type checking dinamico)
 - a tempo di compilazione o "compile time" (type checking statico)

Type mismatch

- Se durante il type checking viene rilevato un mismatch, due sono le strade percorribili:
 - viene segnalato un **errore**
 - viene effettuata una **conversione implicita di tipo**
- Il principio guida delle conversioni implicite è che esse non devono provocare perdita di informazione.

Esempio

- type checking di $E = E1 + E2$
- $\text{tipo}(E1) = \text{tipo}(E2) = \text{int} \rightarrow \text{tipo}(E) = \text{int}$
- $\text{tipo}(E1) = \text{tipo}(E2) = \text{float} \rightarrow \text{tipo}(E) = \text{float}$
- $\text{tipo}(E1) = \text{int}, \text{tipo}(E2) = \text{float} \rightarrow E1$ convertito in reale e $\text{tipo}(E) = \text{float}$
- $\text{tipo}(E1) = \text{float}, \text{tipo}(E2) = \text{int} \rightarrow E2$ convertito in reale e $\text{tipo}(E) = \text{float}$
- Altrimenti: Type Mismatch

Conversione di tipo

- Implicita
- Esplicita: In molti linguaggi esiste un insieme di funzioni che il programmatore può esplicitamente invocare per effettuare conversioni di tipo.
- Ad esempio il **cast** in C/C++:
 - (int)3.4
 - float f=float(i)+4.2

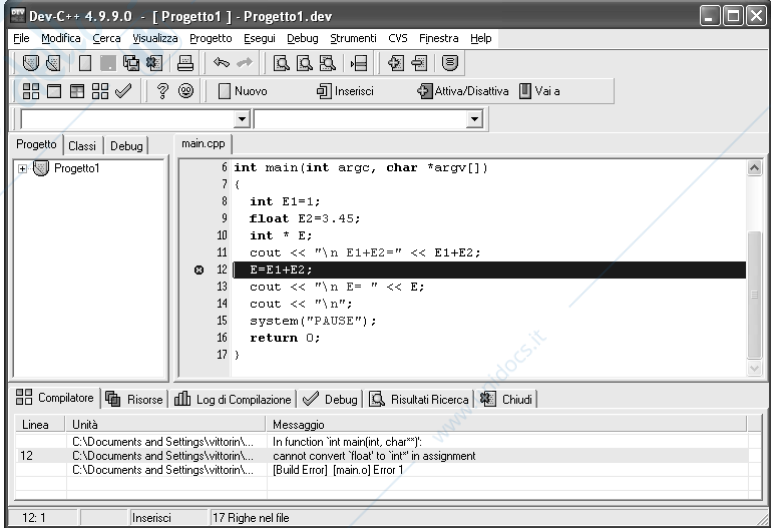
The screenshot shows the Dev-C++ IDE with a C++ program in the main editor. The program calculates the sum of an integer and a float, demonstrating implicit type conversion. The output window shows the result of the calculation. The compiler output window at the bottom displays two warnings related to the type casting.

```

C:\Documents and Settings\vittorin\Documents\UNIVERSITA\IDATTICA\CORSI_LAUREA_LVL...
E1+E2=4.45
E= 4
Premere un tasto per continuare . . .

Dev-C++ 4.9.9.0 - [ Progetto1 ] - Progetto1.dev
File Modifica Cerca Visualizza Progetto Esegui Debug Strumenti CVS Finestra Help
Nuovo Inserisci Attiva/Disattiva Vai a
Progetto Classi Debug [*] main.cpp
Progetto1
6 int main(int argc, char *argv[])
7 {
8     int E1=1;
9     float E2=3.45;
10    int E;
11    cout << "\n E1+E2=" << E1+E2;
12    E=E1+E2;
13    cout << "\n E= " << E;
14    cout << "\n";
15    system("PAUSE");
16    return 0;
17 }

Compilatore Risorse Log di Compilazione Debug Risultati Ricerca Chiudi
Linea Unità Messaggio
12 C:\Documents and Settings\vittorin\... In function 'int main(int, char**)':
12 C:\Documents and Settings\vittorin\... [Warning] assignment to 'int' from 'float'
12 C:\Documents and Settings\vittorin\... [Warning] argument to 'int' from 'float'
13: 25 Modifica Inserisci 17 Righe nel file
  
```



```
Dev-C++ 4.9.9.0 - [ Progetto1 ] - Progetto1.dev
File Modifica Cerca Visualizza Progetto Esegui Debug Strumenti CVS Finestra Help
Nuovo Inserisci Attiva/Disattiva Vai a
Progetto Classi Debug main.cpp
Progetto1
6 int main(int argc, char *argv[])
7 {
8     int E1=1;
9     float E2=3.45;
10    int * E;
11    cout << "\n E1+E2=" << E1+E2;
12    E=E1+E2;
13    cout << "\n E= " << E;
14    cout << "\n";
15    system("PAUSE");
16    return 0;
17 }
Compilatore Risorse Log di Compilazione Debug Risultati Ricerca Chiudi
Linea Unità Messaggio
12 C:\Documents and Settings\vittorin\... In function 'int main(int, char**)':
C:\Documents and Settings\vittorin\... cannot convert 'float' to 'int' in assignment
C:\Documents and Settings\vittorin\... [Build Error] [main.o] Error 1
12:1 Inserisci 17 Righe nel file
```

Riferimenti

- B. Fadini, C. Savy: Fondamenti di Informatica I, Liguori Editore
 - Parte II, cap. 4 par. da 1 a 7