Programmazione Procedurale

Elementi di Base
Lettura e Stampa dei Dati

versione 2.4

1

Sommario

- Introduzione
- Concetto di Flusso (“Stream”)
- Lettura e Stampa da Console in C++
- Stampa su Console in C++
- Lettura da Console in C++
  - ⇨Input Formattato
- Lettura e Stampa da File in C++

2

Introduzione

ATTENZIONE
alla terminologia

- Istruzioni di Stampa
  - ⇨servono a rendere visibili all'esterno del programma i valori delle variabili ("stampare" sui dispositivi di uscita)
- Istruzioni di Lettura
  - ⇨servono ad acquisire dall'esterno i valori delle variabili ("leggere" valori provenienti dai dispositivi di ingresso)

3

Introduzione

- Lettura e stampa da "console"
  - ⇨dati acquisiti dalla tastiera e scritti sullo schermo
  - ⇨utile per l'interazione con l'utente
- Lettura e stampa da file
  - ⇨dati acquisiti dal disco e scritti sul disco
  - ⇨utile per rendere permanenti i risultati delle elaborazioni
- Entrambe basate sullo stesso concetto...

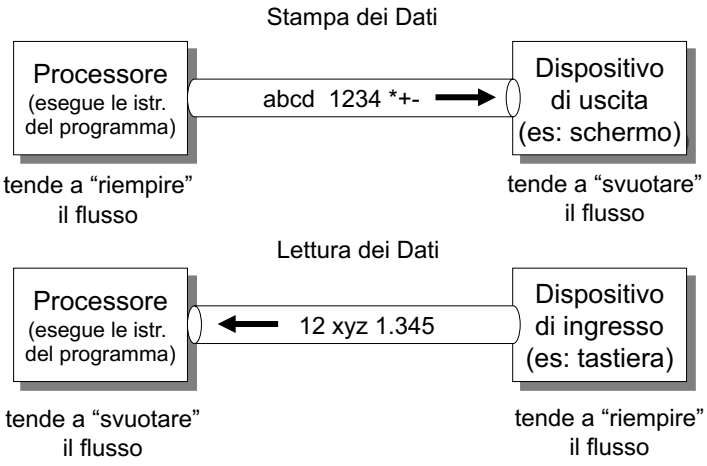
4

Concetto di Flusso

- Flusso (“Stream”)
  - ⇨i dati da stampare vengono visti come una sequenza di caratteri da inviare al dispositivo di uscita (il dispositivo attinge dal flusso)
  - ⇨i dati da leggere vengono visti come provenienti da un flusso che origina dal dispositivo di ingresso (il programma attinge dal flusso)

5

Concetto di Flusso



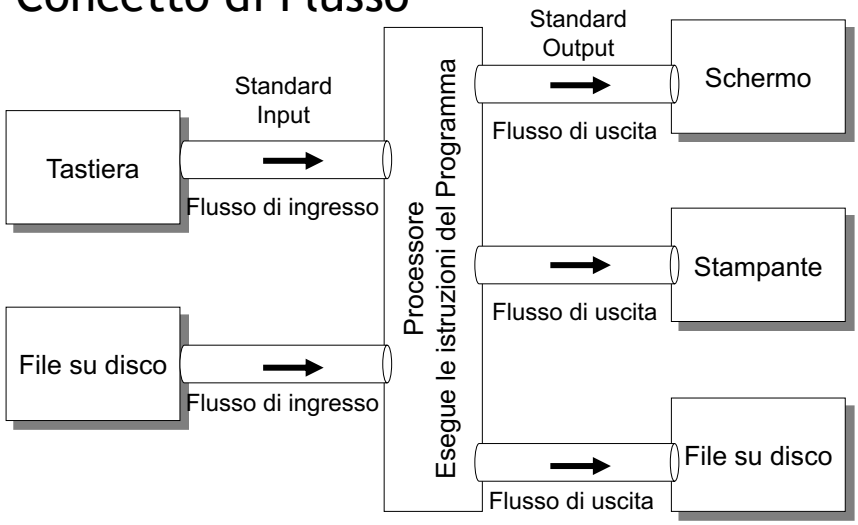
6

Flussi Standard

- I programmi hanno due flussi predefiniti
  - ⇨flusso di uscita: “standard output”
  - ⇨flusso di ingresso: “standard input”
- Normalmente: operazioni su console
  - ⇨“standard output” -> schermo
  - ⇨“standard input” -> tastiera
  - ⇨è possibile redirigere i flussi (es: standard output su stampante)
  - ⇨è possibile definire ulteriori flussi

7

Concetto di Flusso



8

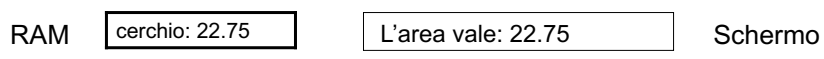
Concetto di Flusso

- Questi concetti
  - ⇨sono comuni a tutti i linguaggi procedurali
  - ⇨ogni linguaggio ha una istruzione di stampa dei dati e una istruzione di lettura dei dati
  - ⇨Es: write e read in Pascal e FORTRAN
  - ⇨In C++ la sintassi è leggermente diversa, (>> e <<)
  - ⇨in tutti i linguaggi la semantica è basata sul concetto di flusso

9

Lettura e Stampa da Console in C++

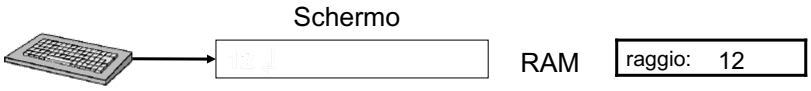
- Stampa sullo schermo
  - ⇨es: `cout << "L'area vale: " << cerchio;`
- Intuitivamente
  - ⇨cout rappresenta il flusso diretto allo schermo ("standard output")
  - ⇨i valori degli argomenti vengono stampati sullo schermo



10

Lettura e Stampa da Console in C++

- Lettura dalla tastiera
  - ⇨es: `cin >> raggio;`
- Intuitivamente
  - ⇨cin rappresenta il flusso proveniente dalla tastiera (“standard input”)
  - ⇨il valore prelevato viene assegnato alla variabile



11

Stampa su Console in C++

- Sintassi

```
cout << <argomento>;
cout << <argomento1> << <argomento2> ... ;
```

- Dove
  - ⇨cout è un oggetto riservato del linguaggio
  - ⇨<< è l’operatore di invio
  - ⇨gli argomenti sono espressioni (costanti, variabili, o espressioni complesse)

12

Stampa su Console

- Semantica
  - ⇨gli argomenti sono considerati da sinistra a destra
  - ⇨viene calcolato il valore di ciascun argomento
  - ⇨il valore viene inserito nel flusso diretto allo schermo sotto forma di sequenza di caratteri
  - ⇨la scheda grafica visualizza il contenuto sullo schermo come sequenza di caratteri

13

Stampa su Console

- Spazio visibile sullo schermo
  - ⇨organizzato in righe
  - ⇨numero di righe virtualmente infinito (scorrimento verticale)
  - ⇨ogni riga è divisa in caratteri
- Cursore
  - ⇨tiene traccia della posizione corrente
  - ⇨inizialmente: primo carattere della prima riga
  - ⇨poi si sposta per via delle stampe

14

Stampa su Console

- Andare a capo
  - ⇨cursore sul primo carattere della riga successiva
  - ⇨equivale a stampare un carattere di fine-riga
  - ⇨il carattere di fine riga è indicato con la parola chiave endl oppure '\n'
- Esempi
  - ⇨`cout << "Vado a capo" << endl;`
  - ⇨`cout << "Vado a capo\n";`

15

Stampa su Console

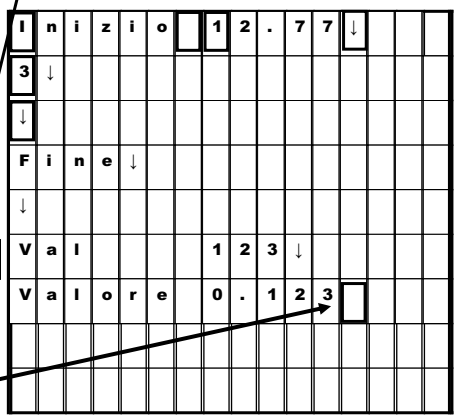
- E' un esempio di "spazio bianco"
- Caratteri per Spazi Bianchi
  - ⇨non producono la stampa di simboli sullo schermo, servono a organizzare le stampe
- In C++
  - ⇨spazio: ' '
  - ⇨fine-riga: endl, '\n'
  - ⇨tabulatore: '\t'

16

Esempi

```
cout << "Inizio";
cout << " ";
cout << cerchio;
cout << endl;
cout << 2+1 << endl;
cout << "\n";
cout << "Fine\n" << '\n';
cout << "Val\t123\n";
cout << "Valore\t0.123";
```

ATTENZIONE
alla differenza



posizione finale
del cursore

17

Lettura da Console in C++

- Sintassi

```
cin >> <variabile>;
cin >> <variabile1> >> <variabile2> ... ;
```

- Dove
  - ⇨cin è un oggetto riservato del linguaggio
  - ⇨>> è l'operatore di prelevamento
  - ⇨gli argomenti sono variabili (devono essere state preliminarmente dichiarate)

18

Lettura da Console

- Semantica
  - ⇨viene effettuato un “input formattato”
  - ⇨su un “flusso tamponato”
- Input formattato
  - ⇨gli “spazi bianchi” vengono trattati in modo particolare (‘ ‘, ‘\n’, ‘\t’)
- Flusso tamponato
  - ⇨il flusso viene analizzato “per righe”

19

Input Formattato

- Flusso tamponato (“buffered”)
  - ⇨i dati digitati dall’utente non sono immediatamente inseriti nel flusso
  - ⇨sono “tamponati” in un’area di memoria temporanea, detta “buffer”
  - ⇨solo quando l’utente digita il carattere di fine riga il contenuto del buffer viene copiato nel flusso
- Inoltre
  - ⇨i caratteri digitati sulla tastiera vengono mostrati sullo schermo (e spostano il cursore)
  - ⇨obiettivo: consentire correzioni

20

Input Formattato

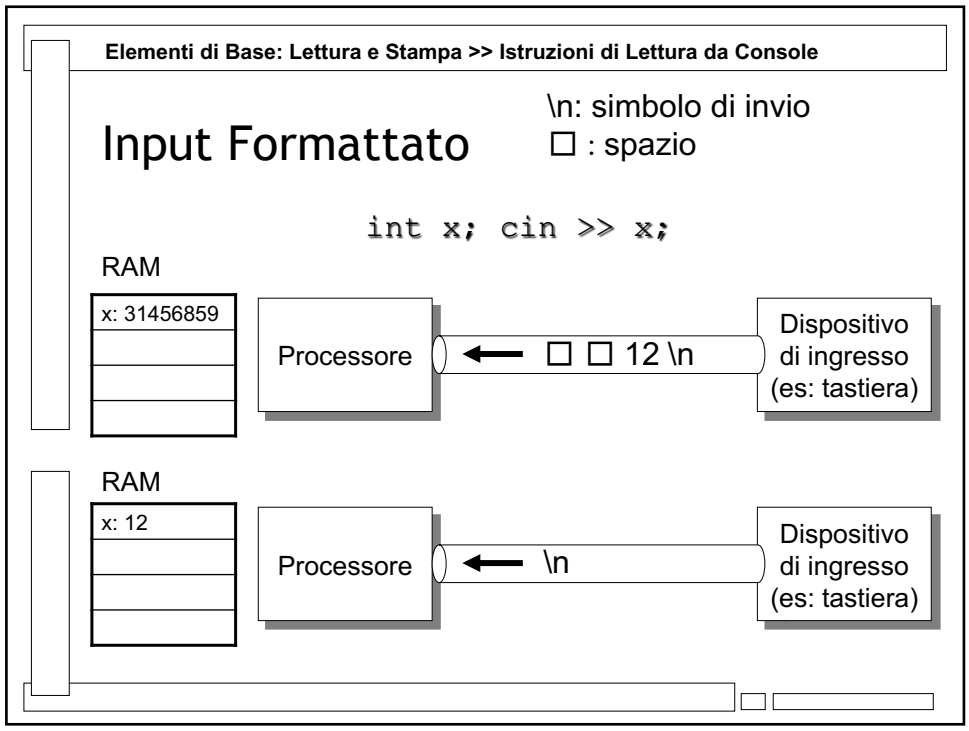
- Input formattato
  - ⇨semantica abbastanza complessa
  - ⇨appena nel flusso è presente almeno una riga, il processore analizza il flusso per "leggere" il valore da assegnare alla variabile
  - ⇨nell'analisi del flusso ignora eventuali spazi bianchi iniziali
  - ⇨gli spazi bianchi vengono utilizzati anche come separatori tra i valori

21

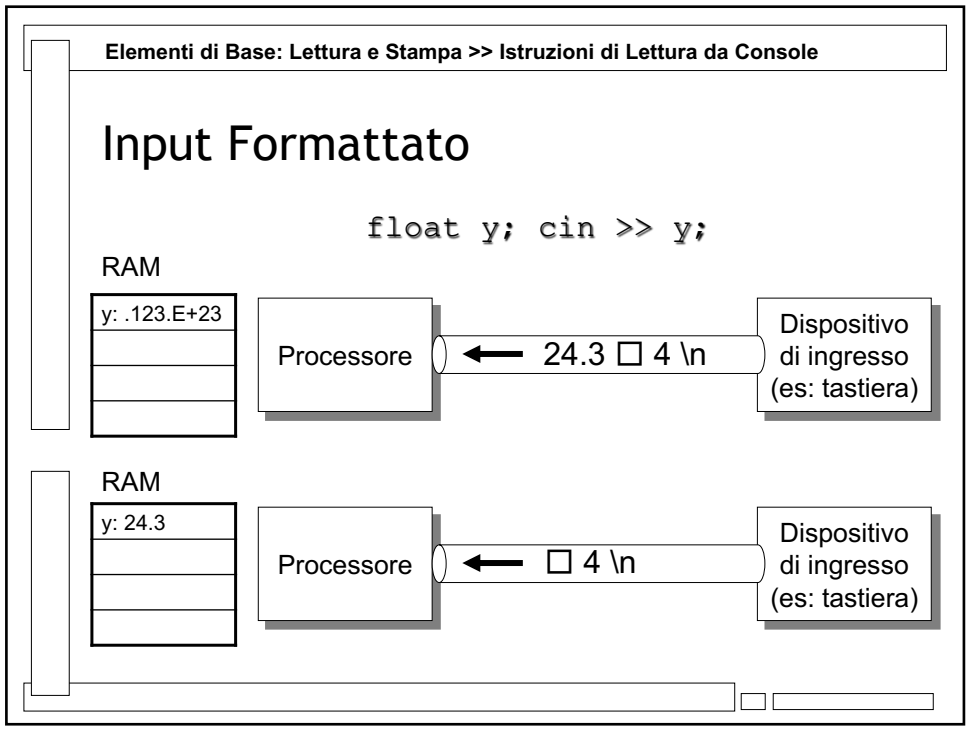
Semantica dell'Input Formattato

- Supponiamo `int x; cin >> x;`
  - ⇨il processore analizza il flusso di ingresso cercando un valore di tipo intero (cioè una sequenza di cifre)
  - ⇨estrae e scarta tutti gli spazi bianchi iniziali che incontra
  - ⇨quando incontra la prima cifra, comincia a leggere il numero considerando tutte le cifre successive
  - ⇨si ferma non appena incontra uno spazio bianco
  - ⇨assegna il numero letto alla variabile x

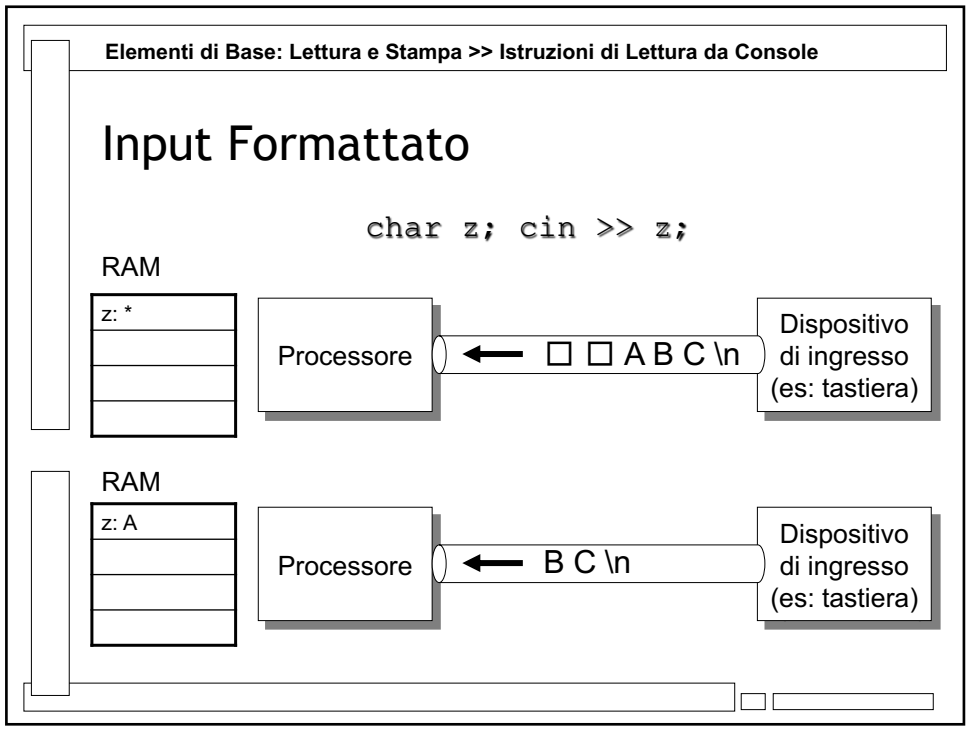
22


Elementi di Base: Lettura e Stampa >> Istruzioni di Lettura da Console
Input Formattato
\n: simbolo di invio
☐ : spazio
int x; cin >> x;
RAM
x: 31456859
Processore
☐ ☐ 12 \n
Dispositivo di ingresso (es: tastiera)
RAM
x: 12
Processore
\n
Dispositivo di ingresso (es: tastiera)

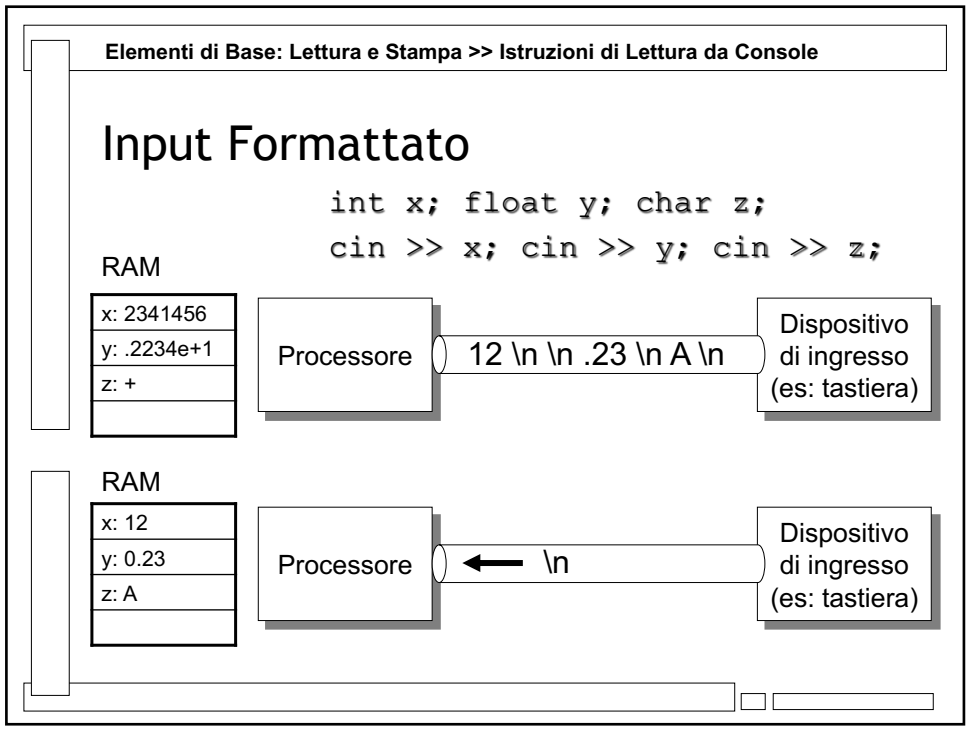
23


Elementi di Base: Lettura e Stampa >> Istruzioni di Lettura da Console
Input Formattato
float y; cin >> y;
RAM
y: .123.E+23
Processore
24.3 ☐ 4 \n
Dispositivo di ingresso (es: tastiera)
RAM
y: 24.3
Processore
☐ 4 \n
Dispositivo di ingresso (es: tastiera)

24


Elementi di Base: Lettura e Stampa >> Istruzioni di Lettura da Console
Input Formattato
char z; cin >> z;
RAM
z: *
Processore
☐ ☐ A B C \n
Dispositivo di ingresso (es: tastiera)
RAM
z: A
Processore
B C \n
Dispositivo di ingresso (es: tastiera)

25


Elementi di Base: Lettura e Stampa >> Istruzioni di Lettura da Console
Input Formattato
int x; float y; char z;
cin >> x; cin >> y; cin >> z;
RAM
x: 2341456
y: .2234e+1
z: +
Processore
12 \n \n .23 \n A \n
Dispositivo di ingresso (es: tastiera)
RAM
x: 12
y: 0.23
z: A
Processore
\n
Dispositivo di ingresso (es: tastiera)

26

Semantica dell'Input Formattato

- Condizioni di errore
  - ⇨si verificano quando la lettura del valore è interrotta da un carattere inaspettato
- Esempio
  - ⇨lettera o simbolo per gli interi
  - ⇨lettera o simbolo (meno il punto '.') per i reali
- Possono verificarsi condizioni inaspettate

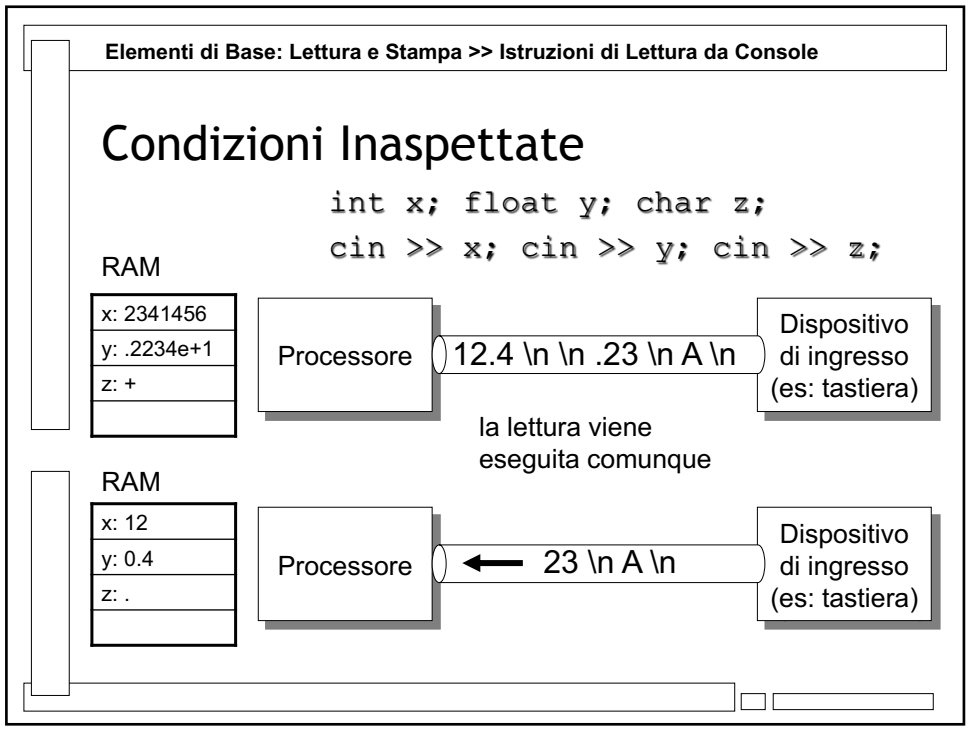
27

Semantica dell'Input Forma

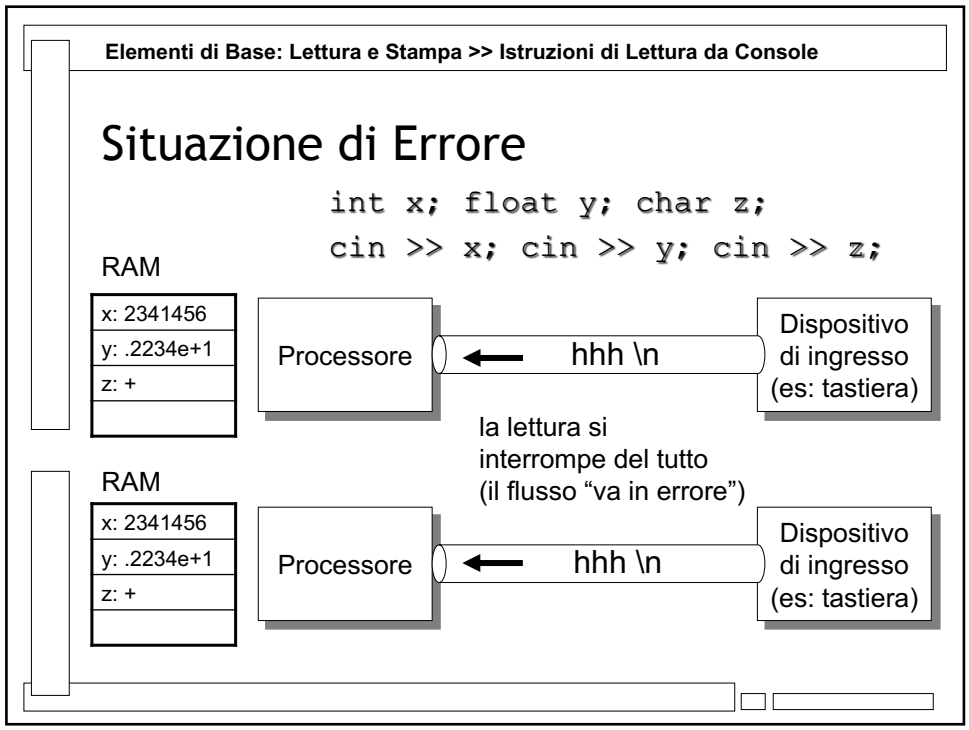
ATTENZIONE
agli errori nelle operazioni di lettura

- Normalmente
  - ⇨il processore cerca di eseguire comunque la lettura producendo risultati scorretti
  - ⇨in altri casi la lettura si interrompe del tutto e i valori delle variabili restano inalterati
- In ogni caso
  - ⇨non viene segnalato un errore esplicito, per cui è necessario fare attenzione

28


Elementi di Base: Lettura e Stampa >> Istruzioni di Lettura da Console
Condizioni Inaspettate
int x; float y; char z;
cin >> x; cin >> y; cin >> z;
RAM
x: 2341456
y: .2234e+1
z: +
Processore
12.4 \n \n .23 \n A \n
Dispositivo di ingresso (es: tastiera)
la lettura viene eseguita comunque
RAM
x: 12
y: 0.4
z: .
Processore
23 \n A \n
Dispositivo di ingresso (es: tastiera)

29


Elementi di Base: Lettura e Stampa >> Istruzioni di Lettura da Console
Situazione di Errore
int x; float y; char z;
cin >> x; cin >> y; cin >> z;
RAM
x: 2341456
y: .2234e+1
z: +
Processore
hhh \n
Dispositivo di ingresso (es: tastiera)
la lettura si interrompe del tutto (il flusso “va in errore”)
RAM
x: 2341456
y: .2234e+1
z: +
Processore
hhh \n
Dispositivo di ingresso (es: tastiera)

30

Lettura e Stampa su File in C++

- I dati prodotti sullo standard output
  - ⇨sono volatili
  - ⇨a volte è necessario salvare permanentemente i dati di un programma
- Memorizzazione permanente su disco
  - ⇨è possibile scrivere dati in un file su disco
  - ⇨è possibile leggere dati da un file su disco

31

Lettura e Stampa su File

- Il meccanismo è del tutto analogo
  - ⇨basato su flussi (“stream”) che vengono associati ai file
- I linguaggi consentono di
  - ⇨creare nuovi flussi oltre a quelli standard associandoli a file sul disco
  - ⇨utilizzare le istruzioni ordinarie di lettura e stampa per lavorare con questi flussi

32

Lettura e Stampa su File

- In C++
  - ofstream: flusso associato ad un file in cui si stampano dei dati

    es: `ofstream flussoSuFile ("fileDati.txt");`
  - ifstream: flusso associato ad un file da cui si leggono dei dati

    es: `ifstream flussoDaFile ("fileDati.txt");`
- Per utilizzarli è necessario specificare
  - `#include <fstream>`

33

Stampa su File

- Sintassi
  - dichiarazione di flusso di stampa su file

    `ofstream <nomeFlusso> ("<nomeFileSuDisco>");`
- Dove
  - <nomeFlusso> è un identificatore
  - <nomeFileSuDisco> è il nome fisico di un file sul disco
- Esempi

```
ofstream flussoDaFile ("fileDati.txt");
ofstream altroFlusso ("c:\prova.txt");
```

34

Stampa su File

ATTENZIONE
all'apertura e alla chiusura dei flussi su file

- Semantica
  - ⇨viene creato un nuovo flusso associato al file
  - ⇨se il file non esiste viene creato
  - ⇨se esiste, il suo contenuto viene cancellato
- Da quel momento
  - ⇨il flusso creato può essere utilizzato esattamente come cout (operatore <<)
  - ⇨nel file vengono scritti caratteri (file di testo)
- ATTENZIONE
  - ⇨al termine delle operazioni il flusso deve essere chiuso

35

Stampa su File: Esempio

```
#include <iostream>
#include <fstream>
using namespace std;
int main() {
      int x;
      float y;
      ofstream flussoSuDisco ("prova.txt");
      cout << "Immetti due numeri" << endl;
      cin >> x;
      cin >> y;
      flussoSuDisco << x << " " << y << endl;
      flussoSuDisco.close();
      return 0;
}
```

>> fileUscita.cpp

36

Flusso di Lettura da File

- Sintassi
  - ⇨dichiarazione di un flusso di lettura da file

```
ifstream <nomeFlusso> ("<nomeFileSuDisco>");
```

- Dove
  - ⇨<nomeFlusso> è un identificatore
  - ⇨<nomeFileSuDisco> è il nome fisico di un file sul disco
- Esempi

```
ifstream flussoDaFile ("fileDati.txt");
ifstream altroFlusso ("c:\prova.txt");
```

37

Lettura da File

- Semantica
  - ⇨viene creato un nuovo flusso di lettura associato al file
- Da quel momento
  - ⇨il flusso creato può essere utilizzato esattamente come cin (operatore >>)
- Attenzione
  - ⇨il file deve esistere su disco
  - ⇨altrimenti comportamenti scorretti
  - ⇨anche in questo caso il flusso deve essere chiuso

38

Lettura da File: Esempio

```
#include <iostream>
#include <fstream>
using namespace std;
int main() {
      int x;
      float y;
      ifstream flussoDiLettura ("prova.txt");
      flussoDiLettura >> x;
      flussoDiLettura >> y;
      cout << "I due numeri sono" << endl;
      cout << x << " " << y;
      flussoDiLettura.close();
      return 0;
}
```

>> fileIngresso.cpp

39

Riassumendo

- Concetto di Flusso (“Stream”)
- Stampa su Console
- Lettura da Console
  - ⇨input formattato (ATTENZIONE)
- Lettura e Stampa da File
  - ⇨#include <fstream>
  - ⇨creazione di un flusso di stampa su file
  - ⇨creazione di un flusso di lettura da file

40

Termini della Licenza

41