

---

# Ingresso e Uscita Variabili e costanti intere

# Operatore di uscita <<

---

- Scrittura formattata su *cout*
- *cout<<obj1<<obj2<<...<<endl ;*
- Il generico oggetto da stampare può anche essere una *variabile* o una *costante* con nome

# Riepilogo iostream

---

- Ogni file sorgente che contenga riferimenti ad oggetti della libreria di ingresso/uscita deve contenere le direttive

```
#include <iostream>  
using namespace std;
```

- Devono precedere il primo punto in cui viene utilizzato uno stream di ingresso/uscita

# Riepilogo primo esercizio

---

```
#include <iostream>  
  
using namespace std;  
  
main()  
  
{  
  
    cout<<"Ciao mondo!\n" ;  
  
}
```

# Riepilogo secondo esercizio

---

```
#include <iostream>
```

```
using namespace std;
```

```
main()
```

```
{
```

```
    cout<<"Ciao mondo!"<<endl ;
```

```
}
```

# Riepilogo compilazione

---

- Sintassi più semplice per generare un programma eseguibile da un file sorgente:
- `g++ nome_sorgente.cc`
  - Assegna un nome predefinito al programma eseguibile (a.out)
- `g++ -o nome_eseguibile nome_sorgente.cc`
  - Permette di scegliere il nome del programma eseguibile

# Esercizio 1/2

---

- Scrivere un programma in cui si definisce una variabile intera e se ne stampa il valore sullo schermo

# Esercizio 2/2

---

```
#include <iostream>  
using namespace std;  
  
main()  
  
{  
  
    int i = 10 ;  
  
    cout<<i;  
  
}
```

# Esercizio 1/2

---

- Scrivere un programma in cui si definisce una variabile intera e se ne stampa il valore sullo schermo col seguente formato:

*Il valore della variabile è 10.*

- E si va a capo

# Esercizio 2/2

---

```
#include <iostream>  
using namespace std;  
  
main()  
  
{  
  
    int i = 10 ;  
  
        cout<<"Il valore della variabile è  
        "<<i<<". "<<endl ;  
  
}
```

# Una soluzione alternativa

---

```
#include <iostream>  
using namespace std;  
  
main()  
  
{  
  
int i = 10 ;  
  
cout<<"Il valore della variabile è "<<i ;  
  
cout<<". "<<endl ;  
  
}
```

# Ritorno valore finale

---

- Un processo ritorna un valore quando termina
- Può essere letto ed utilizzato dal programma che lo ha invocato
- La funzione *main* può (in effetti dovrebbe sempre) essere dichiarata ritornare un valore di tipo intero
- Il valore di ritorno della funzione *main* è il valore ritornato dal processo stesso

# Esempio

---

```
#include <iostream>
using namespace std;

int main()
{
    int i = 10 ;

    cout<<"Il valore della variabile è "<<i ;
    cout<<". "<<endl;

    return 0 ;

}
```

# Esercizio 1/2

---

- Scrivere un programma in cui si definisce una variabile intera e se ne stampa il valore sullo schermo, quindi se ne cambia il valore e si stampa di nuovo il (nuovo) valore sullo schermo

# Esercizio 2/2

---

```
#include <iostream>  
using namespace std;
```

```
main()
```

```
{
```

```
int i = 10 ;
```

```
cout<<"Il valore della variabile è "<<i<<endl ;
```

```
i = 12 ;
```

```
cout<<"Il nuovo valore è "<<i<<endl ;
```

```
}
```

# Esercizio 1/2

---

- Scrivere un programma in cui si definisce una costante intera e se ne stampa il valore sullo schermo col seguente formato:

*Il valore della costante è 10.*

- E si va a capo

# Esercizio 2/2

---

```
#include <iostream>  
using namespace std;  
  
main()  
  
{  
  
    const int i = 10 ;  
  
    cout<<"Il valore è "<<i<<". "<<endl ;  
  
}
```

# Lettura dallo *stdin*

---

- Operatore di ingresso `>>` applicato ad un oggetto di tipo *istream*
- *Esempio: cin>>nome\_variabile ;*
- Legge i caratteri in ingresso dallo *standard input* (abbreviato *stdin*)
- Li interpreta in base al tipo della variabile
- Assegna il valore letto alla variabile di nome *nome\_variabile*

# Esempio di interpretazione

---

- `cin >> a;`
- La variabile `a` è di tipo `int`
- Se l'utente scrive `23` e va a capo, Si leggono i caratteri `2`, `3` e `\n`
- Vengono interpretato come le due cifre decimali del numero `23`
- Il numero `23` viene memorizzato nella variabile `a`

# Esercizio 1/2

---

- Si scriva un programma che legge un valore intero da tastiera e lo stampa a video

# Esercizio 2/2

---

```
#include <iostream>  
using namespace std;  
  
main()  
  
{  
  
    int i ;  
  
    cin>>i ;  
  
    cout<<"Il valore inserito è "<<i<<endl ;  
  
}
```

# Ingresso inconsistente

---

- Cosa accade se la sequenza di caratteri letta non rappresenta alcun numero in notazione decimale?
- **Il valore del secondo argomento rimane invariato** (non avviene alcuna memorizzazione)
- **Le successive letture falliranno**
- Vedremo in futuro come resettare lo stato dello stream per non fare più fallire le successive letture

# Esercizio 1/2

---

- Miglioriamo l'esercizio precedente
- Vogliamo stampare anche un messaggio di richiesta del numero da inserire:

*Inserisci un valore intero: 13*

*Il valore inserito è: 13*

- L'operatore di ingresso `>>` applicato al *cin* **non scrive sullo standard output (stdout)**

# Esercizio 2/2

---

```
#include <iostream>  
using namespace std;  
  
main()  
  
{  
  
int i ;  
  
cout<<"Inserisci un valore intero " ;  
  
cin>>i ;  
  
cout<<"Il valore inserito è "<<i<<endl ;  
  
}
```

# Esercizio 1/3

---

- Scrivere un programma che legge in ingresso due valori interi e stampa il risultato della moltiplicazione tra i due numeri

*Inserisci il primo numero: 10*

*Inserisci il secondo numero: 20*

*10 \* 20 = 200*

- Calcolare, usando il vostro programma, il valore di  $19312 * 7284$

# Esercizio 2/3

---

```
#include <iostream>
using namespace std;
main()
{
    int i, j, ris ;

    cout<<"Inserisci il primo numero " ;
    cin>>i ;
    cout<<"Inserisci il secondo numero " ;
    cin>>j ;

    ris = i * j ;
    cout<<i<<"*"<<j<<" = "<<ris<<endl ;
}
```

# Esercizio 3/3

```
#include <iostream> /* Soluzione alternativa: */
using namespace std; /* senza variabile di */
main() /* appoggio */
{
    int i, j;

    cout<<"Inserisci il primo numero " ;
    cin>>i ;
    cout<<"Inserisci il secondo numero " ;
    cin>>j ;

    cout<<i<<"*"<<j<<" = "<<i*j<<endl ;
}
```

# Esercizio 1/3

---

- Scrivere un programma che legge in ingresso due valori interi e stampa sia il risultato della divisione intera tra i due numeri che il resto della divisione stessa

*Inserisci il primo numero: 5*

*Inserisci il secondo numero: 2*

*5 / 2 = 2 con resto 1*

- Calcolare, usando il vostro programma, il valore della divisione intera e del resto di  
 $19312 / 7284$

# Esercizio 2/3

---

```
#include <iostream>
using namespace std;
main()
{
    int i, j, div, resto ;

    cout<<"Inserisci il primo numero " ;
    cin>>i ;
    cout<<"Inserisci il secondo numero " ;
    cin>>j ;

    div = i / j ;
    resto = i % j ;
    cout<<i<<" / "<<j<<" = "<<div<<" con resto
    "<<resto<<endl ;
}
```

# Esercizio 3/3

```
#include <iostream>      /* Soluzione alternativa */
using namespace std;    /* senza variabili di */
main()                  /* appoggio */
{
    int i, j ;

    cout<<"Inserisci il primo numero " ;
    cin>>i ;
    cout<<"Inserisci il secondo numero " ;
    cin>>j ;

    cout<<i<<" / "<<j<<" = "<<i/j<<" con resto
        "<<i%j<<endl ;
}
```

# Esercizio 1/4

---

- Scrivere un programma che legge in ingresso due valori interi e li memorizza in due variabili, quindi scambia il contenuto delle variabili e lo stampa sullo schermo

*Inserisci il valore di i: 2*

*Inserisci il valore di j: 3*

*Dopo lo scambio:  $i = 3, j = 2$*

# Esercizio 2/4

---

- Primo esercizio un po' più difficile
- Riflettiamo un po' sul problema: se assegniamo  $i$  a  $j$  abbiamo perso il valore di  $i$  e viceversa ...
- Sfruttiamo questo esercizio per capire in generale come **acquisire la mentalità giusta**

# Sviluppo di una soluzione

---

- Un buon ordine con cui arrivare a risolvere un problema nuovo di cui non si conosce la soluzione è il seguente:
  - 1) Riflettere sul problema finché non si è sicuri di aver capito a sufficienza tutti gli aspetti e le implicazioni
  - 2) Cercare di farsi venire un'idea che sembri buona per risolvere il problema (o almeno per partire)
  - 3) Provare a definire l'algoritmo e controllarlo per capire se è corretto (eventualmente modificarlo)
  - 4) Quando si è sicuri dell'algoritmo, partire con la codifica
  - 5) Collaudare il programma per verificare che faccia veramente quello che deve

# Commenti

---

- A meno di problemi molto molto semplici, non rispettare il precedente ordine porta quasi sempre a risultati mediocri o pessimi
- Il tipico errore che si commette è quello di incominciare a scrivere il programma prima di aver chiaro l'algoritmo (se non addirittura il problema stesso)
- Il passo 2 può essere quello più critico, perché richiede un atto creativo in mancanza del quale non si sa da dove partire
- La capacità di compiere con successo tale passo si accresce con l'esercizio alla soluzione dei problemi e l'esperienza

# Proviamo ...

---

- ... ad applicare le precedenti fasi dello sviluppo al nostro problema dello scambio di variabili
- Ci vuole innanzitutto un'idea ...

- Se memorizziamo il valore di una delle due variabili, per esempio di  $i$ , in una terza variabile d'appoggio, allora, quando assegnamo il valore di  $j$  ad  $i$ , non abbiamo perso il valore di  $i$  !
- Possiamo quindi assegnare a  $j$  il (precedente) valore di  $i$ , salvato nella variabile di appoggio

# Algoritmo

---

- 1)Assegnare il valore contenuto in **i** ad una variabile d'appoggio **app**
- 2)Assegnare il valore contenuto in **j** ad **i**
- 3)Assegnare a **j** il valore contenuto nella variabile di appoggio (uguale al valore che **i** aveva prima del passo 2)

Se l'algoritmo ci è chiaro e ci sembra corretto, non ci resta che provare ad implementarlo ...

# Programma 1/2

---

```
#include <iostream>  
using namespace std;
```

```
int main()
```

```
{
```

```
    int i, j ;
```

```
    cout<<"Inserisci il valore di i " ;
```

```
    cin>>i ;
```

```
    cout<<"Inserisci il valore di j " ;
```

```
    cin>>j ;
```

# Programma 2/2

---

```
int appoggio = i ;
```

```
i = j ;
```

```
j = appoggio ;
```

```
cout<<"Dopo lo scambio: i = "<<i
```

```
<<" , j = "<<j ;
```

```
return 0 ;
```

```
}
```

# Esercizio più difficile

---

- Scrivere un programma che legge in ingresso due valori interi e li memorizza in due variabili, quindi scambia il contenuto delle variabili e lo stampa sullo schermo
- Ma senza utilizzare **nessuna** variabile d'appoggio!
- *scambia\_senza\_appoggio.cc*

# Altri dettagli sui processi

---

- Ad ogni processo è associato un identificatore numerico: *pid*
- Per elencare i propri processi:

*ps x*

- Per elencare tutti i processi:

*ps ax*

# Ancora sui segnali

---

- **Ctrl + C** manda il segnale SIGINT (interruzione) al processo in esecuzione
- A volte tale segnale può non bastare per interrompere il processo
- Comando *kill*: spedisce segnali ai processi. Per uccidere il processo:  
*kill -9 pid*
- Invocato da un altro terminale

# Compiti per casa 1

---

- Scrivere i seguenti programmi
  - rispettando le fasi di sviluppo precedentemente viste
  - senza utilizzare istruzioni di controllo di flusso (niente istruzioni condizionali ed iterative, ma solo esecuzione sequenziale)
  - facendo uso **solo** di variabili di tipo *int* e dei relativi operatori
    - $+$ ,  $-$ ,  $*$ ,  $/$ ,  $\%$ , `abs()`
    - *Data una variabile intera  $i$ , l'operatore `abs(i)` ritorna il valore assoluto di  $i$*

- Su alcune macchine, per usare la funzione `abs()` bisogna aggiungere la direttiva

```
#include <stdlib.h>
```

all'inizio del programma

# Compiti per casa 2

---

- Scrivere un programma che legge in ingresso un numero intero, lo interpreta come un tempo espresso in secondi, e lo stampa in minuti e secondi (*da\_sec\_a\_min\_sec.cc*)

*Tempo in secondi? 67*

*Equivalgono a 1 min, 7 sec*

# Compiti per casa 3

---

- Scrivere un programma che legge in ingresso due numeri, li interpreta come un tempo espresso in minuti e secondi, e lo stampa in secondi (*da\_min\_sec\_a\_sec.cc*)
  - **Attenzione:** per semplicità assumiamo come valido anche un ingresso in cui il secondo numero sia maggiore di 59

*Minuti ? 3*

*Secondi ? 78*

*Equivalgono a 258 secondi*

# Compiti per casa 4

---

- Scrivere un programma che legge in ingresso quattro numeri, li interpreta come due tempi espressi in minuti e secondi, e stampa la differenza tra i due tempi, espressa in secondi (*soluzione non fornita*)
  - **Attenzione:** per semplicità assumiamo come valido anche un ingresso in cui il secondo numero sia maggiore di 59

*Minuti e secondo primo tempo ? 3 45*

*Minuti e secondi secondo tempo ? 5 36*

*Differenza: 111*

# Compiti per casa 5

---

- Scrivere un programma che legge in ingresso quattro numeri, li interpreta come due tempi espressi in minuti e secondi, e stampa la differenza tra i due tempi, di nuovo espressa in minuti e secondi (*soluzione non fornita*)
- **Attenzione:** per semplicità assumiamo come valido anche un ingresso in cui il secondo numero sia maggiore di 59

*Minuti e secondo primo tempo ? 3 45*

*Minuti e secondi secondo tempo ? 5 36*

*Differenza: 1 51*

# Compiti per casa 6

---

- Scrivere un programma che legge in ingresso un numero intero e stampa 0 se il numero è pari, 1 altrimenti (*0\_se\_pari.cc*)

*Inserisci un numero intero: 23*

*1*

- Scrivere un programma che legge in ingresso un numero intero e stampa 1 se il numero è pari, 0 altrimenti (*1\_se\_pari.cc*)

# Compiti per casa 7

---

- Scrivere un programma che legge in ingresso due numeri interi positivi, poi stampa 0 se il primo è multiplo dell'altro, 1 altrimenti (*0\_se\_multiplo.cc*)

*Inserisci il primo numero intero positivo: 32*

*Inserisci il secondo numero intero positivo: 11*

*1*

# Compiti per casa 8

---

- Scrivere un programma che legge in ingresso due numeri interi positivi, poi stampa 1 se il primo è multiplo dell'altro, 0 altrimenti (*1\_se\_multiplo.cc*)

*Inserisci il primo numero intero positivo: 32*

*Inserisci il secondo numero intero positivo: 11*

*0*

# Compiti per casa 9

---

- Scrivere un programma che legge in ingresso un numero intero diverso da 0, e stampa -1 se è negativo, 1 se è positivo (*1\_se\_pos-1\_se\_neg.cc*)

*Inserisci un numero intero: -3*

*-1*

# Compiti per casa 10

---

- Scrivere un programma che legge in ingresso un numero intero diverso da 0, e stampa 0 se è negativo, 1 se è positivo (*0\_se\_neg\_1\_se\_pos.cc*)

*Inserisci un numero intero: -3*

*0*

# Compiti per casa 11

---

- Scrivere un programma che legge in ingresso un numero intero diverso da 0, e stampa 1 se è negativo, 0 se è positivo

*Inserisci un numero intero: -3*

*1*