

---

Ingresso e Uscita  
Variabili e costanti intere  
Processo risolutivo

# Riepilogo sintassi

---

- 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

# Stampa variabili

---

- Stessa sintassi che si utilizza per stampare ad esempio una stringa
- Solo che stavolta si passa il nome della variabile

# Esercizio 1/2

---

- Scrivere un programma in cui si definisce una variabile intera inizializzata con il valore che si preferisce, 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 e' 10.*

- E si va a capo

# Esercizio 2/2

---

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

# Una soluzione alternativa

---

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

# 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
- Utilizzare un numero maggiore di 255 come nuovo valore

# Esercizio 2/2

---

```
#include <iostream>  
using namespace std;  
  
main()  
  
{  
  
    int i = 10 ;  
  
    cout<<"Il valore della variabile è "<<i<<endl ;  
  
    i = 1229 ;  
  
    cout<<"Il nuovo valore è "<<i<<endl ;  
  
}
```

# Esercizio 1/2

---

- Si scriva un programma che legge un valore intero da tastiera e lo stampa a video
- Provare anche ad immettere un numero maggiore di 255

# Esercizio 2/2

---

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

# 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*

# 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 (sfida nella prossima slide)

*Inserisci il primo numero: 5*

*Inserisci il secondo numero: 2*

*5 / 2 = 2 con resto 1*

# 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 ...
- Fermiamoci un attimo e sfruttiamo questo semplice esercizio per fare un importantissimo passo per la nostra preparazione
  - **acquisire la mentalità giusta** per realizzare, partendo da un problema, un programma che lo risolva con successo

# Sviluppo di una soluzione

---

- Un buon ordine con cui arrivare a risolvere, mediante un programma, 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

# Proviamo ...

---

- ... ad applicare le precedenti fasi dello sviluppo al nostro problema dello scambio di variabili
- Dopo esserci assicurati di aver veramente capito il problema, ci vorrà un'idea ...
- Prima di discutere una possibile idea, provate da soli
- Probabilmente vedremo un primissimo esempio delle conseguenze del mancato rispetto delle precedenti fasi

# Soluzione sbagliata 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 ;
    cout<<"i = "<<j<<"", j = "<<i<<endl ;
```

# Soluzione sbagliata 2/2

---

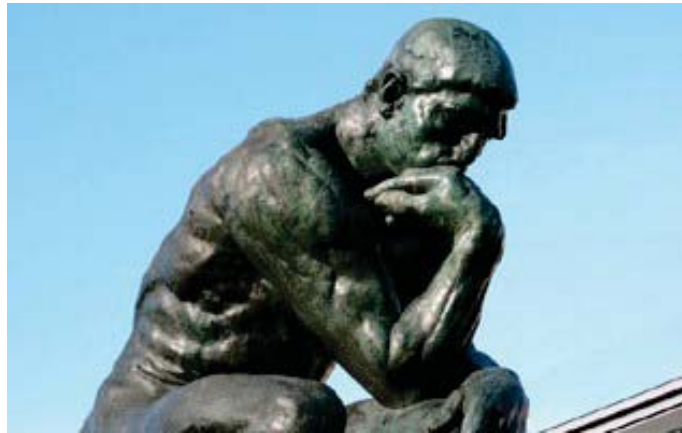
- Abbiamo semplicemente stampato il contenuto della variabili in ordine invertito
- Non abbiamo scambiato il contenuto delle variabili come richiesto
- Perché siamo andati così fuori strada?

# Errore al primo passo

---

- Perché non abbiamo eseguito il primo passo con la massima cura possibile: non abbiamo riflettuto abbastanza sul problema
- Se vogliamo, possiamo utilizzare questa lezione per migliorarci molto per il futuro
- Se invece preserviamo questo approccio superficiale il nostro livello di risolutori di problemi rimarrà abbastanza basso
- Chiarito il primo passo, vediamo assieme un'idea per arrivare ad una soluzione





- 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 ;  
}
```