



Linguaggio C++

marco.falda@unipd.it



Uso del compilatore

Il comando per il compilatore GNU C++ ha la seguente sintassi di base:

```
g++ -o file_out file_in
```

Se si usano funzioni matematiche:

```
g++ -lm -o file_out file_in
```



Input / Output

- In C++ *standard* si usa la libreria **iostream**
 1. `#include <iostream>`
 2. `using namespace std;`
 3. “`cout <<`” per l'*output*,
“`cin >>`” per l'*input*

Esempio

```
#include <iostream>  
int main()  
{  
    using namespace std;  
    int n;  
    cout<<"Inserire un numero:";  
    cin>>n;  
    cout<<"Hai inserito " <<n;  
    return 0;  
}
```



Tipi di dato e variabili

- In C++ esistono pochi tipi di dato fondamentali:
 - `bool`, `int`, `long`, `float`, `double`, `char` (, `void`)
- Tutte le variabili devono essere dichiarate prima dell'uso
 - possono essere assegnate con “=”

Operatori matematici

- Operazioni usuali (+, -, *, /);
- Operatore di resto intero (%);
- + e - si trovano anche nella forma ++ e -- (pre/post in/decremento):

`cout << $\pm\pm$ a; \Leftrightarrow`

`a = a \pm 1; cout << a;`

`cout << a $\pm\pm$; \Leftrightarrow`

`cout << a; a = a \pm 1;`

Operatori di confronto

- Esistono gli operatori comuni: $<$, $>$, $>=$ e $<=$.
- Uguaglianza e disuguaglianza sono rappresentate da: $==$ e $!=$.
- Esiste un unico operatore ternario detto “*if aritmetico*”:
(espressione)? vera: falsa;

– esempio: $c = (a > b)? a : b;$

Operatori logici

- Sono operatori che restituiscono “vero” o “falso”:
 - && (AND) è vero se entrambi gli operandi sono veri;
 - || (OR) è vero se almeno uno degli operandi è vero;
 - ! (NOT) è vero se il suo operando è falso.

Selezioni - if

```
if (espr.) {  
    ...  
}  
else if (espr.) {  
    ...  
} ... // else if  
else {  
    ...  
}
```

Selezioni - switch

```
switch (espr.) {  
  case cost1:  
    ...; break;  
  case costN:  
    ...; break;  
  default:  
    ...  
}
```



Esercizio 1

I/O, variabili

- Scrivere un programma che acquisisce in ingresso due numeri interi maggiori di zero e ne calcola il resto intero



Esercizio 2

Operatori logici, if

- Scrivere un programma che acquisisce in ingresso un numero intero che rappresenta un anno e determina se tale anno è bisestile
 - un anno è bisestile se è divisibile per 4 ma non per 100 altrimenti non lo è, con l'eccezione di quelli che sono divisibili per 100 e 400



Esercizio 3

Operatori di confronto, switch

- Scrivere un programma che acquisisce in ingresso tre numeri interi indicanti giorno mese ed anno di una data e stampa la data equivalente espressa come `<gg.mese.aaaa>`
 - ad esempio 22 8 2008 → 22 agosto 2008
- Devono essere rifiutate le terne che costituiscono date non valide

Uso del “debugger”

- È un programma utile per rintracciare gli errori logici
- In Linux compilare aggiungendo “-g” e avviare con `kdbg ./nomefile &`
 - esempio:
`kdbg ./calcolatrice &`

