

# Linguaggio C++

marco.falda@unipd.it

1

## 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
```

2

## 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*

3

## Esempio

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

4

## 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 "="

5

## Operatori matematici

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

`cout << ±±a;`       $\Leftrightarrow$       `a = a ± 1; cout << a;`

`cout << a±±;`       $\Leftrightarrow$       `cout << a; a = a ± 1;`

6

## 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;`

7

## 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.

8

## Selezioni - *if*

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

9

## Selezioni - *switch*

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

10

---

## Esercizio 1

### I/O, variabili

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

---

11

---

## 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 per 400

---

12

## 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

13

## 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 &`

14

---

## Uso del “debugger”

- Per tracciare un programma
    1. porre un “breakpoint”
    2. eseguire il programma
    3. continuare in modo “passo-passo”
  
  - Si possono ispezionare le variabili (*locals*) e le chiamate delle funzioni (*stack*)
-