



UNIVERSITÀ DEGLI STUDI DI PADOVA

Istruzioni, dichiarazioni, definizioni (*ioni)

Stefano Ghidoni



Agenda

- Istruzioni
- Dichiarazioni e definizioni
- Inizializzazione



- Un'istruzione o statement è una parte di codice C++ che specifica un'azione
 - **Non sono istruzioni le direttive del preprocessore (es., #include)**
- Termina con ;
 - Necessario per interpretare il codice
- Vari tipi di istruzioni
 - Qualche esempio?



Tipi di istruzioni

- Expression statements
 - L'istruzione è la valutazione dell'espressione
- Dichiarazioni
- Selezione (if-statement, switch-statement)
- Iterazione (while-statement, for-statement)
- ...

Istruzioni per gestire le
variabili (e non solo):
dichiarazioni e definizioni



- Dichiarazione: istruzione che **introduce un nome in uno scope**
 - Specifica il nome
 - Specifica il tipo
 - Opzionalmente, specifica un inizializzatore (valore iniziale)
- Un nome deve essere dichiarato prima di essere usato
- Dichiarazione di:
 - Variabili
 - Funzioni
 - ... (molto altro)

es: creando il main e una variabile in esso, questa esiste fino alla fine del main; è la porzione di codice in cui la variabile è stata dichiarata.



- Esempi di dichiarazioni – per ciascuna delle seguenti, verificate:
 - Specifica il nome?
 - Specifica il tipo?
 - Opzionalmente, specifica un iniziatore (valore iniziale)?

```
int a = 7;  
const double cd = 8.7;  
double sqrt (double);           // una funzione con un  
                                 // argomento double  
vector<Token> v;                 // vettore di variabili Token
```

→ il tipo è "vettore di Token"



- Una dichiarazione ha effetto sulla memoria?

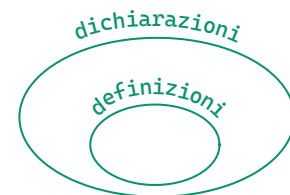
```
int a = 7;  
const double cd = 8.7;  
double sqrt (double);           // una funzione con un  
                                // argomento double  
vector<Token> v;                 // vettore di variabili Token
```

Una dichiarazione è "quando un nome entra in circolo".

Nel terzo esempio, la dichiarazione della funzione, non vi sono effetti sulla memoria. Quindi non ha sempre effetto sulla memoria.



- Definizione: una dichiarazione che specifica completamente l'entità dichiarata
 - Una definizione è anche una dichiarazione



```
int a = 7;  
vector<double> v;  
double sqrt (double d) { /* ... */ }
```

- Alcune dichiarazioni *non* sono definizioni

```
double sqrt (double);  
extern int a;
```

Esiste una variabile "a" dichiarata in un altro file, dove le è stata riservata la memoria.
In questo modo viene riconosciuto il simbolo ma non gli viene riservata altra memoria.
Non è consigliato usarlo



- Una definizione:
 - "Fornisce tutte le informazioni necessarie per creare l'entità nella sua interezza"
 - Può avere un significato diverso a seconda dell'entità definita:
 - **Variabile:** fornisce tipo e nome, riserva la memoria
 - **Funzione:** fornisce il corpo della funzione
 - **Classe:** fornisce variabili e funzioni membri della classe
- Quando un'entità è definita, per forza è anche dichiarata



Dichiarazioni vs definizioni

- In alcuni contesti (**non in questo corso**), dichiarazione e definizione sono considerati mutuamente esclusivi
 - Dichiarazione intesa come *dichiarazione che non è anche una definizione*