

Classi di memorizzazione delle variabili in C++

Versione semplificata di materiale didattico messo in rete da un docente di Unina di cui ignoro il nome.

In ogni caso il link al file originale è [questo](#)

Variabili automatiche

- Sono le variabili locali ad un blocco di istruzioni.
- Sono definite in un blocco di istruzioni e sono allocate e deallocate con esso.
- La visibilità lessicale è locale al blocco.
- Nell'immagine in memoria del processo (cioè del programma in esecuzione), sono allocate nell'area *stack*.

Immagine di un processo in memoria

Esempio:

Istruzioni	{	Area codice
Dati		Area dati globale
	{	Area <i>heap</i>
		Area stack x

```
{  
  int x; // var. autom.  
  ...  
}
```

Variabili esterne

- Sono le variabili globali e vengono definite all'esterno di ogni blocco.
- La visibilità lessicale si estende a tutto il programma, ma per renderle visibili anche da procedure contenute in file diversi, devono essere ivi dichiarate con la parola chiave *extern*.
- La loro estensione temporale si estende a tutto il programma (sono allocate all'inizio del programma per essere poi deallocate alla fine dello stesso).
- In memoria vengono allocate nell'area dati statici o globale.

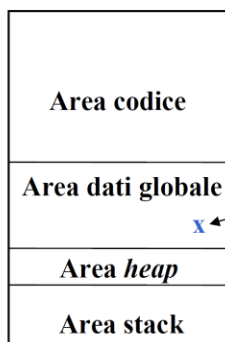
Esempio:

Area codice
Area dati globale x
Area <i>heap</i>
Area stack

```
int x; // var. esterna (globale)  
  
void main () {  
  ...  
}
```

Variabili automatiche statiche

- Sono variabili automatiche prefissate con la parola chiave *static*.
- La visibilità lessicale è locale al blocco in cui sono definite.
- La loro estensione temporale si estende a tutto il programma (sono allocate all'inizio del programma per essere poi deallocate alla fine dello stesso).
- In memoria vengono allocate nell'area dati globale.

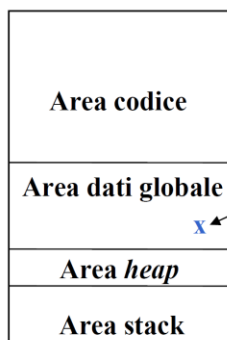


Esempio:

```
void f() {  
    static int x; // var. autom. statica  
    ...  
}
```

Variabili esterne statiche

- Sono variabili esterne prefissate con la parola chiave *static*.
- La visibilità lessicale è locale al file in cui sono definite.
- La loro estensione temporale si estende a tutto il programma (sono allocate all'inizio del programma per essere poi deallocate alla fine dello stesso).
- In memoria vengono allocate nell'area dati globale.



Esempio:

```
static int x; // var. esterna statica  
  
void main () {  
    ...  
}
```

Variabili dinamiche

- Sono definite durante l'esecuzione del programma mediante *puntatori*.
- Sono accessibili dovunque esiste un riferimento ad esse.
- L'allocazione e la deallocazione sono controllate dal programmatore.
- In memoria vengono allocate nell'area *heap*.

Quadro riepilogativo

Classe di memorizzaz.	Definizione	Visibilità	Durata	Allocazione
Automatiche	Definite in un blocco	Blocco	Blocco	Area <i>stack</i>
Esterne	Definite all'esterno di ogni blocco	Tutto il programma	Tutto il programma	Area dati statici
Automatiche statiche	Definite in un blocco + <i>static</i>	Blocco	Tutto il processo	Area dati statici
Esterne statiche	Definite all'esterno di ogni blocco + <i>static</i>	All'interno del file in cui sono definite	Tutto il processo	Area dati statici
Dinamiche	Tramite puntatori e operatori <i>new/delete</i>	Finché esiste un riferimento valido	Definita dal programmatore	Area <i>heap</i>