



UNIVERSITÀ  
degli STUDI  
di CATANIA

# Variabili, tipi primitivi, operazioni aritmetiche in C++

Corso di programmazione I AA 2020/21

Corso di Laurea Triennale in Informatica

---

Prof. Giovanni Maria Farinella

Web: <http://www.dmi.unict.it/farinella>

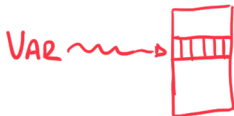
Email: [gfarinella@dm.unict.it](mailto:gfarinella@dm.unict.it)

Dipartimento di Matematica e Informatica

1. Variabili, costanti e commenti in C++
2. Sistema di numerazione in base 2
3. Rappresentazione dei numeri al calcolatore: standard IEEE 754
4. Tipi in C++
5. Operatori aritmetici, funzioni matematiche di base, conversioni

# Variabili, costanti e commenti in C++

---



Cosa è una variabile?

*Un contenitore di dati* identificato da un **nome** all'interno del programma.

La variabile corrisponde ad un certo **indirizzo nella memoria del calcolatore** nel quale si esegue il programma.

Una variabile è associata a un certo **tipo**, che deve essere **adeguato** a rappresentare l'informazione che si vuole memorizzare all'interno della variabile.

# Variabili in C++

## Definizione di variabile

```
int numero_di_ordini = 10; // COMMENTO
```

Handwritten annotations in red:

- A bracket above `int` is labeled "tipo".
- A bracket under `numero_di_ordini` is labeled "NOME".
- A bracket under `= 10` is labeled "INIZIALIZZAZIONE".
- A bracket under the semicolon `;` is labeled "FINE".



Usare un nome che “descriva” il contenuto della variabile.

**Inizializzare la variabile?** Il C++ non lo rende obbligatorio ma è spesso necessario per evitare errori.

Il carattere “;” termina la singola istruzione che definisce la variabile.

Definizione senza inizializzazione:

```
int j; ←
```

Definisce una variabile int e la inizializza con il valore 10:

```
→ int numero_di_ordini = 10;
```

Il contenuto della `totale_ordini_euro` viene inizializzato con il valore attuale di alcune variabili:

```
int totale_ordini_euro = prezzo * quantita;
```

# Variabili in C++

Errore in fase di compilazione!

```
int codice = "AA10";
```

Definisce una o più variabili contemporaneamente:

```
int v1, v2=3, v3;
```

NB: v1 e v3 nessun valore iniziale, v2 inizializzata!

```
int v2;  
v2=3;  
int v3;
```

Definisce una variabile di tipo carattere (e la inizializza):

```
char c = 'a';
```

## Valori letterali e variabili

**Inizializziamo** le variabili con espressioni che usano **letterali** e/o **altre variabili**.

Un valore **letterale** è un elemento del programma che rappresenta un valore.

true, 1.0, 40 sono letterali di tipo, rispettivamente booleano, double, e intero.

"acqua" è un valore letterale di tipo stringa, mentre 'c' è un letterale di tipo char.



# Variabili in C++

Letterale	Tipo	Note
-6	int	int non ha parte frazionaria, può essere negativo
0.5	double	Viene rappresentato in memoria come un double
0.5f	float	Viene rappresentato in memoria come un float
1E6	double	Notazione esponenziale. Equivale a $1 \times 10^6$ oppure 1000000
10,456	#	<b>Errore</b> in fase di compilazione! Va usato il punto, non la virgola..
3 1/2	#	<b>Errore</b> in fase di compilazione! Va usata una espressione in forma decimale: <u>3.5</u>

## Regole per i nomi delle variabili

- nomi debbono **iniziare** con una lettera, oppure underscore (" \_"); *\_Var Var \_Var\_alterna*
- i **rimanenti** caratteri possono essere anche numeri, oppure ancora lettere o underscore.
- • **NO spazi** nel nome delle variabili!
- ATTENZIONE: C/C++ **case-sensitive**!
  - var e Var non sono la stessa variabile.
- Le parole *riservate* (e.g. `double`) non si possono usare per i nomi di variabile..

*int Var;*

# Variabili in C++

$a = a + 1;$        $b = a++;$        $b = ++a;$        $\left. \begin{array}{l} b \mapsto a \\ b \mapsto a+1 \end{array} \right\} \begin{array}{l} a \\ a+1 \\ a+1 \end{array}$

In che modo può cambiare il valore di una variabile nel tempo?

- Assegnamento: ES:  $a=10;$ ,  $b=z-20;$
- Incremento/decremento: (forma postfissa)  $a++;$   
 $b--;$ , (forma prefissa)  $++a;$   $--b;$
- Istruzione di input:  $\text{cin} >> x;$

Usare la stessa variabile a destra e sinistra di un assegnamento.

Cosa succede nel seguente assegnamento?

```
var = var + 10;
```

1. **Leggi** il contenuto della variabile `var`
2. **Somma**, al valore letto in precedenza, il valore 10
3. **Copia** il valore ottenuto dalla precedente somma nella variabile `var`

## Ordine di inizializzazione delle variabili

Il seguente codice è (semanticamente) corretto?

```
1 int a=10;  
2 int b;  
3 int volume = a * b * altezza;  
4 b=15;
```

Attenzione alla inizializzazione (tardiva) di b.

Quale sarà il valore di b nella valutazione della espressione alla riga 3?

```
const int valore_banconota_A = 10;
```

Si vuole assegnare un nome ad uno o più valori **costanti**.

La parola riservata const per una variabile permette di indicare al compilatore che

- il valore di tale variabile **const non può cambiare** rispetto al suo valore iniziale.
- quella variabile **const** va **inizializzata in fase di creazione**.



Le costanti migliorano la **leggibilità**, quindi la comprensione del codice e permettono di ridurre gli errori in fase di codifica.

### Esempio

```
(A) int somma_iniziale = num_banconote * 10;
```

VS

```
(B) int somma_iniziale = num_banconote * VALORE_BANCONOTA;
```

## Esempio

```
int somma_iniziale = num_banconote * 10;
```

VS

```
(B) int somma_iniziale = num_banconote * VALORE_BANCONOTA;
```

E se il programmatore avesse la necessità di cambiare il valore corrispondente a `valore_banconota` da 10 a 20?

Con (A) può farlo cambiando una singola riga di istruzione, mentre con (B)...



# Variabili costanti

Il compilatore **ignora** il testo che rappresenta un commento

## Commento singola linea

→ float alpha = 0.5; // deve essere < beta

## Commento multi-linea

→ /\*

Questo programma calcola il profitto medio:

- mensile

- annuo

→ \*/

int main(){...}

## Esempi svolti

A1\_00\_var.cpp