

# C++

Manuale per le ripetizioni di Informatica

Marco Lampis

6 dicembre 2022

# Indice

<b>0</b>	<b>Informazioni generali</b>	<b>1</b>
0.1	Come svolgo le lezioni . . . . .	1
<b>1</b>	<b>if</b>	<b>3</b>
1.1	Sintassi . . . . .	3
1.1.1	Esempio . . . . .	3
1.1.2	Esercitazione . . . . .	4
<b>2</b>	<b>For</b>	<b>5</b>
2.1	Sintassi . . . . .	5
2.1.1	Esempio . . . . .	5
2.2	Cicli annidati . . . . .	5
2.2.1	Esempio . . . . .	6
2.3	Esercitazione . . . . .	6
<b>3</b>	<b>Strutture (struct)</b>	<b>9</b>
3.1	Sintassi . . . . .	9
3.2	Accedere ai membri . . . . .	9
3.3	Esercitazione . . . . .	10



# 0 Informazioni generali

Ciao! Sono **Marco**, sono uno studente magistrale in *Software Engineering* al Politecnico di Torino e mi sono laureato in Ingegneria Informatica all'università di Pisa. Nella vita sono un programmatore, uno smanettone e amante di videogiochi! Amo quello che studio, e per questo motivo fornisco ripetizioni di informatica con particolare attenzione a:

- programmazione (Java, C++, C, Python, C#, Javascript, PHP)
- algoritmi e strutture dati
- basi di Dati
- più o meno tutto quello che riguarda l'informatica!

Sia per studenti delle scuole superiori che per l'università.

## 0.1 Come svolgo le lezioni

La mia metodologia è innovativa e interattiva: utilizzo una piattaforma apposita per lavorare contemporaneamente sullo stesso file con gli studenti, come se fossimo accanto. In questo modo, le lezioni sono più coinvolgenti e divertenti, e gli studenti possono imparare in modo facile ed efficiente. Inoltre, le lezioni includono sia esercitazioni che approfondimenti teorici, aiuto compiti e revisione.

Per aiutare gli studenti a prepararsi al meglio e a esercitarsi anche a casa, ho creato un sito web dedicato con risorse utili come teoria, esercizi e slide. Su questo sito, gli studenti possono trovare tutto il materiale di cui hanno bisogno, e il materiale è sempre disponibile gratuitamente per i miei studenti. Inoltre, uso un iPad per prendere appunti durante le lezioni e poi condividerli con gli studenti, in modo che possano rivedere tutto ciò che è stato trattato.

Se volete migliorare le vostre conoscenze in informatica e avere un supporto personalizzato e professionale, non esitate a contattarmi. Sarò felice di fornirvi maggiori informazioni e di fissare una lezione con voi.



# 1 if

Il costrutto **if** consente di verificare se una condizione è verificata ed eseguire un blocco di codice in base al risultato. L'esito è sempre di tipo **booleano**, ovvero può essere solo **true** o **false**.

## 1.1 Sintassi

La sintassi di **if** è la seguente:

```
1  if (condizione1){  
2      // codice eseguito se la condizione é vera  
3  }  
4  
5  else if(condizione2){  
6      // codice eseguito se la condizione1 non é vera  
7      // ma é vera condizione2  
8  }  
9  
10 else{  
11     // codice eseguito se nessuna delle condizioni é vera  
12 }
```

**Attenzione:** Il blocco di codice che segue **else if** ed **else** è opzionale. Se non viene specificato nessun blocco di codice, il programma non esegue alcuna operazione. Inoltre è importante scrivere le keyword con caratteri minuscoli, in caso contrario si avrà errore

**Suggerimento:** Non utilizzare **else if** ed **else** se non è necessario, in quanto possono rendere il codice meno leggibile.

### 1.1.1 Esempio

```
1  int a = 5;  
2  int b = 6;  
3
```

```
4  if (b > a){
5      cout<<"b é maggiore di a";
6  }
7
8  else if (a == b){
9      cout<<"a e b sono uguali";
10 }
11
12 else{
13     cout<<"a é maggiore di b";
14 }
```

Puoi trovare a questo link alcuni esercizi su **if**: <https://esercizi.mlampis.dev/book.php?langs=cpp>

### 1.1.2 Esercitazione

Verificare se un numero inserito in input è pari.

```
1  #include <iostream>
2  using namespace std;
3
4
5  int main(){
6
7      // dichiariamo una variabile
8      int a;
9
10     // invitiamo l'utente a inserire un valore
11     cout<<"Inserisci un numero: ";
12
13     // lo facciamo inserire
14     cin >> a;
15
16     // verifichiamo se un numero é pari
17     if (a%2 == 0){
18         cout<<"Numero pari";
19     }
20
21     else {
22         cout<<"Numero dispari";
23     }
24
25     return 0;
26 }
```

## 2 For

Il ciclo **for** viene utilizzato quando si conosce il numero di iterazioni da compiere. Il ciclo **for** è composto da tre parti:

- **inizializzazione**: viene inizializzato un contatore
- **verifica**: viene verificata la condizione di uscita dal ciclo
- **passo**: viene incrementato il contatore

### 2.1 Sintassi

La sintassi di **for** è la seguente:

```
1 for (inizializzazione; verifica; passo){  
2   // codice eseguito  
3 }
```

**Nota:** spesso si utilizza il contatore **i** per indicare il numero di iterazioni.

#### 2.1.1 Esempio

Esempio di esercizio che stampa i numeri da 1 a 5 (escluso):

```
1 for (int i = 0; i < 5; i++) {  
2   cout << i << "\n";  
3 }
```

### 2.2 Cicli annidati

E' possibile inserire un **for** all'interno di un'altro **for** (senza limiti di volte), ottenendo un ciclo annidato. In questo caso, il ciclo esterno viene eseguito prima del ciclo interno.



### 2.2.1 Esempio

```
1 // ciclo esterno
2 for (int i = 1; i <= 2; ++i) {
3     cout << "Outer: " << i << "\n"; // Eseguito 2 volte
4
5     // Inner loop
6     for (int j = 1; j <= 3; ++j) {
7         cout << " Inner: " << j << "\n"; // Eseguito 6 volte (2 * 3)
8     }
9 }
```

## 2.3 Esercitazione

Dato un numero mostrato a schermo, mostrarne la tabellina.

**esempio di output:**

```
1 Quale tabellina vorresti mostrare? 5
2 5 * 1    = 5
3 5 * 2    = 10
4 5 * 3    = 15
5 5 * 4    = 20
6 5 * 5    = 25
7 5 * 6    = 30
8 5 * 7    = 35
9 5 * 8    = 40
10 5 * 9    = 45
11 5 * 10   = 50
```

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     // creiamo la variabile per salvare la tabellina
7     int numero;
8
9     cout<<"Quale tabellina vorresti mostrare? ";
10
11     // facciamo inserire all'utente il numero
12     cin>>numero;
13
14     // vogliamo fare una azione ripetuta
15     // partena -> 1
16     // arrivo -> 10 (incluso)
17     for(int cont=1; cont <= 10; cont = cont+1){
```

```
18
19 // stiamo mostrando qualcosa a schermo:
20 // numero " * " cont " = " prodotto
21 int prodotto= numero*cont;
22
23 // 5 * 1 = 5
24 cout<<numero<<" * "<<cont<<" = "<<prodotto<<endl;
25 }
26
27 return 0;
28 }
```



## 3 Strutture (struct)

Le strutture, solitamente chiamate `struct`, consentono il raggruppamento di più informazioni all'interno di un solo contenitore.

Puoi pensare a una struct come a una scatola con scompartimenti, in cui ogni scompartimento può contenere a sua volta altri oggetti.

### 3.1 Sintassi

Per realizzare una struttura è sufficiente utilizzare la keyword `struct` seguita dalle `{}`; al loro interno andremo a mettere le variabili che faranno parte della nostra struttura.

```
1 struct {           // dichiarazione della struttura
2     int contenuto1; // membro (variabile int)
3     char contenuto2; // membro (variabile char)
4     // ...
5 } nomeStruttura;    // nome con cui verrà richiamata
```

### 3.2 Accedere ai membri

Per accedere a una variabile membro della struttura si utilizza il nome della variabile seguita dal carattere `..`.

```
1 // dichiarazione della struttura
2 struct {
3     int contenuto1; // membro (variabile int)
4     char contenuto2; // membro (variabile char)
5     // ...
6 } nomeStruttura;    // nome con cui verrà richiamata
7
8 // istanzio una struttura
9 nomeStruttura st1;
10
11 // ne assegno i valori
```

```
12 st1.contenuto1 = 1;  
13 st1.contenuto2 = 'A';
```

### 3.3 Esercitazione

Costruire un record **Prodotto** con:

- nome
- marca
- prezzo unitario
- data scadenza
- quantità in magazzino

Fare l'input e l'output di un prodotto, stampare poi il valore totale del prodotto accantonato in magazzino.

```
1 #include <stdio.h>  
2 #include <stdlib.h>  
3  
4 // definiamo il valore massimo di caratteri in una stringa  
5 #define size 20  
6  
7 // creazione della struttura per l'inserimento della data  
8 struct data{  
9     int giorno;  
10    int mese;  
11    int anno;  
12 };  
13  
14 // creazione della struttura per il prodotto  
15 struct prodotto{  
16     char nome[size];  
17     char marca[size];  
18     float prezzo_unitario;  
19     struct data scadenza;  
20     int quantita;  
21 };  
22  
23  
24 int main(){  
25  
26     // creiamo una variabile per salvare  
27     // le informazioni del prodotto  
28     struct prodotto pippo;  
29  
30     // chiediamo di inserire il nome
```

```
31 printf("Inserire il nome del prodotto: ");
32 scanf("%s", pippo.nome);
33
34 // chiediamo di inserire la marca
35 printf("Inserire la marca del prodotto: ");
36 scanf("%s", pippo.marca);
37
38 // chiediamo di inserire il prezzo unitario
39 printf("Inserire il prezzo unitario: ");
40 scanf("%f", &pippo.prezzo_unitario);
41
42 // chiediamo la scadenza del prodotto
43 printf("Inserire il giorno di scadenza: ");
44 scanf("%d", &pippo.scadenza.giorno);
45
46 printf("Inserire il mese di scadenza: ");
47 scanf("%d", &pippo.scadenza.mese);
48
49 printf("Inserire l'anno di scadenza: ");
50 scanf("%d", &pippo.scadenza.anno);
51
52 // inseriamo la quantità dei prodotti
53 printf("Inserire la quantità di prodotto: ");
54 scanf("%d", &pippo.quantita);
55
56
57 printf("Il prodotto %s ha le seguenti caratteristiche: \n", pippo.
    nome);
58 printf("- nome: %s\n", pippo.nome);
59 printf("- marca: %s\n", pippo.marca);
60 printf("- prezzo unitario: %f\n", pippo.prezzo_unitario);
61 printf("- scadenza: %d-%d-%d\n", pippo.scadenza.anno, pippo.scadenza.
    mese, pippo.scadenza.giorno);
62 printf("- quantita': %d\n", pippo.quantita);
63
64 float valore_totale;
65 valore_totale = pippo.prezzo_unitario* pippo.quantita;
66
67 printf("Il valore totale e' di %.2f euro", valore_totale);
68
69 return 0;
70 }
```

