



Laboratorio di Programmazione 1

**Docente: dr Damiano Macedonio
Lezione 7 - 7/11/2013**

1

Original work Copyright © Sara Migliorini,
University of Verona

Modifications Copyright © Damiano Macedonio,
University of Verona

Istruzione While

```
while( espressione ){  
    istruzione_1;  
    ...  
    istruzione_n;  
}
```

- Ad ogni iterazione viene valutato il valore di espressione:
 - Se il risultato è *vero* allora viene eseguito il corpo del ciclo.
 - Se il risultato è *falso* allora il ciclo termina e si continua con l'istruzione immediatamente successiva al ciclo.
- Se nessuna istruzione all'interno del corpo del ciclo modifica la valutazione di `espressione`, si ha un ciclo infinito.

Ciclo For o Ciclo While?

- Tutti i cicli **for** possono essere tradotti in un corrispondente ciclo **while**:

```
for( espr_iniziale; cond_ciclo; espr_ciclo ){  
    istruzione_1;  
    ...  
    istruzione_n;  
}
```

```
espr_iniziale;  
while( cond_ciclo ){  
    istruzione_1;  
    ...  
    istruzione_n;  
    espr_ciclo;  
}
```

Ciclo For o Ciclo While?

- E viceversa... Tutti i cicli `while` possono essere tradotti in un corrispondente ciclo `for`:

```
while( espressione ){  
    istruzione_1;  
    ...  
    istruzione_n;  
}
```

```
for( ; espressione; ){  
    istruzione_1;  
    ...  
    istruzione_n;  
}
```

Ciclo For o Ciclo While?

- La scelta tra ciclo for o ciclo while dipende dalle caratteristiche dell'iterazione che si deve eseguire.
- Il ciclo for è preferibile quando:
 - Il ciclo deve essere eseguito un numero di volte fisso e noto prima di iniziare il ciclo.
 - Se l'espressione iniziale, l'espressione del ciclo e la condizione del ciclo riguardano tutte la stessa variabile.

Istruzione Do-While

- I cicli `for` e `while` eseguono il controllo della condizione prima che il ciclo venga eseguito.
 - Il corpo del ciclo potrebbe non essere mai eseguito.
- Il ciclo `do-while` valuta la condizione alla fine.
 - Garantisce che il corpo del ciclo venga eseguito almeno una volta.

```
do {  
    istruzione_1;  
    ...  
    istruzione_n;  
} while( espressione );
```

Istruzione Do-While

- Esecuzione del ciclo `do-while`:
 1. Viene eseguito il corpo del ciclo.
 2. Viene valutata l'espressione del ciclo.
 - Se il valore dell'espressione è vero il corpo del ciclo viene eseguito nuovamente.
 - Se il valore dell'espressione è falso il ciclo termina e si continua con l'istruzione immediatamente successiva.

Esercizio 4

- Scrivere un programma C che inverte le cifre di un numero intero digitato dall'utente.
- Per estrarre la cifra più a destra da un numero intero si può prendere il resto dell'intero diviso 10: $1234 \% 10 = 4$.

Esercizio 5

- Modificare il programma dell'esercizio precedente usando un ciclo do-while invece che while, in modo che anche la cifra 0 sia invertita.

Esercizio 6

- Si scriva un programma che calcoli la somma di una sequenza di numeri interi inserita dall'utente. La lunghezza della sequenza non è impostata a priori, ma termina quando l'utente inserisce il numero zero.

Esercizio 7

- Si supponga di partecipare a una festa, per socializzare si stringono le mani a tutti i partecipanti. Si scriva un programma che calcola il numero totale di strette di mano.
Suggerimento: quando una persona arriva, stringe la mano a chi è già presente. Si può usare un ciclo per individuare il numero di strette di mano effettuate ogni volta che arriva una nuova persona.

Esercizio 8

- Si scriva un programma che chieda all'utente di inserire l'altezza di un triangolo isoscele.
- Si visualizzi il triangolo utilizzando righe di asterischi.
- La base del triangolo è *verticale*, il vertice è rivolto *a sinistra*.
- *In pratica*. La prima riga avrà un asterisco solo, la seconda due e così via. Ciascuna riga avrà un asterisco in più della precedente fino a raggiungere l'altezza indicata dall'utente. Per le righe successive, il numero di asterischi per riga deve decrescere di uno per ogni nuova riga.

Esercizio 8

Esempio di Output

Altezza 5

```
*  
**  
***  
****  
*****  
****  
***  
**  
*
```

Esercizio 9

- Come nell'esercizio 8, chiedere all'utente di inserire l'altezza di un triangolo isoscele.
- Si visualizzi il triangolo utilizzando righe di asterischi.
- La base del triangolo è *orizzontale*, il vertice è rivolto in *alto*.

Esercizio 9

Esempio di Output

Altezza 5:

```
  *
 * * *
 * * * * *
 * * * * * * *
 * * * * * * * * *
```

Altezza 7:

```
      *
     * * *
    * * * * *
   * * * * * * *
  * * * * * * * * *
 * * * * * * * * * *
* * * * * * * * * * *
```


Esercizio 10

- Come nell'esercizio 9, chiedere all'utente di inserire l'altezza di un triangolo isoscele (l'altezza deve essere almeno 2).
- Utilizzare gli asterischi per delimitare solo il perimetro del triangolo.
- Anche in questo caso, la base del triangolo è *orizzontale*, il vertice è rivolto in *alto*.
- Per uniformità la base deve essere disegnata intervallando spazi e asterischi.

Esercizio 10

Esempio di Output

Altezza 5:

```
  *
 * *
 *   *
 *     *
* * * * *
```

Altezza 7:

```
  *
 * *
 *   *
 *     *
 *       *
 *         *
 *           *
* * * * * *
```