Loops

Loops

- sequence of instructions that is continually repeated until a certain condition is reached.

Loops

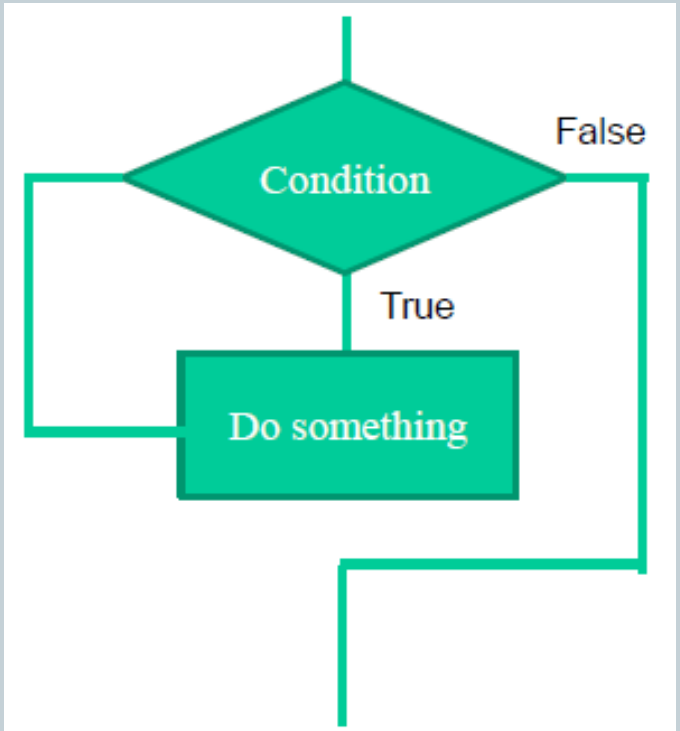
Pseudocode:

Loop

execute instruction(s)

Until Condition

Flowchart:



Loops

- There are three types which are common to most programming languages:
  - Condition Tested Loops
  - Counted Loops
  - End less Loops

Condition Tested Loops

- repeats a set of instructions until a certain condition is reached.
- The condition can be tested at the start of the loop (before any of the instructions are executed), during the loop, or at the end of the loop.

Counted Loops

- It allows programmer to instruct the computer to perform a set of instructions ***n*** times,
- usually ***n*** is an integer value

Counted Loops

- There are generally two ways that the number of repetitions of a loop will be know ahead of time:
  - The loop always repeats the same number of times.
  - The program calculates the number of repetitions based upon user input.

Endless Loops

- An endless loop goes round and round until one of three things happens:
  - The computer is turned off (or the application stopped, forcefully)
  - The computer encounters an EXIT(or similar)statement
  - An error forces the application to 'crash'
- Some endless loops serve a purpose, in message loops
  - Example: where it is necessary to continually monitor for incoming messages from the operating system.

Example of Loop Statement

- loop statement in programming language
  - FOR Loop
  - WHILE Loop
  - DO…WHILE Loop

FOR Loop

- A FOR loop is a loop that repeats a specified number of times.
- The loop uses a counter to tell it how many times to run the same sequence of activities.

FOR Loop

- The counter has the following three numeric values:
  - Initial counter value
  - Increment (the amount to add to the counter each time the loop runs)
  - Final counter value
- The loop ends when the counter reaches the final counter value, or , if there is an associated test condition, when the test condition is true.

FOR Loop

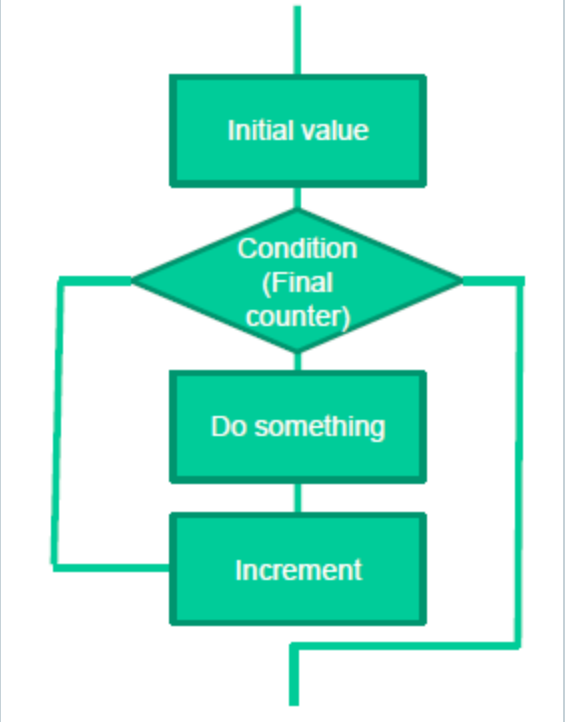
Pseudocode:

FOR ***n*** *times*

execute instruction(s)

Increment

Flowchart:



FOR loop syntax:

FOR(initial counter value, final counter, increment)

execute instruction(s)

FOR loop In C

```
for ( int intCounter = 1; intCounter <= 10 ;
                              intCounter ++ )  // count from 1 to x
{
    printf( " counter  increment   " ); // output ' '
 } //end for loop
```

OR

```
int intCounter ;
 for ( intCounter = 1; intCounter <= 10 ;
                    intCounter ++ )  // count from 1 to x
 {
    printf( " counter  increment   " ); // output ' '
 } //end for loop
```

FOR Loop

```
 int intCounter  ;
for (intCounter = 5;  counter  > 0 ;
                    counter -- )  // count from 1 to x
      {
          printf( " counter  decrement   " ); // output ' '
       } //end for loop
```

WHILE Loop

- loop that repeats while some condition is satisfied.
- tests its' condition at the beginning of every loop.
- If the condition is false at the start, the sequence of activities contained in the loop never runs at all.

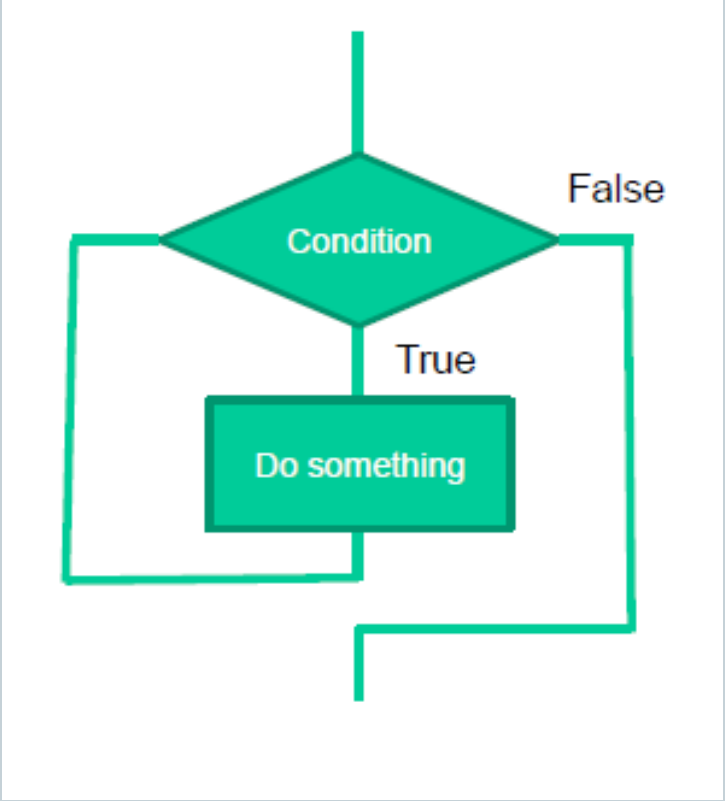
WHILE Loop

Pseudocode:

WHILE *condition*

execute instruction(s)

Flowchart:



WHILE loop syntax:

WHILE(Condition)

execute instruction(s)

WHILE loop in C

```
int intCounter = 1; // initialization


while (intCounter <= 10) //repetition condition
{
printf ( "%d\n", counter ); // display counter
intCounter = intCounter +1 ;
// increment (or counter ++)
}
// end while
```

DO-WHILE Loop

- Like a while loop, a do-while loop is a loop that repeats while some condition is satisfied.
- Unlike a while loop, a do-while loop tests its condition at the end of the loop.

DO-WHILE Loop

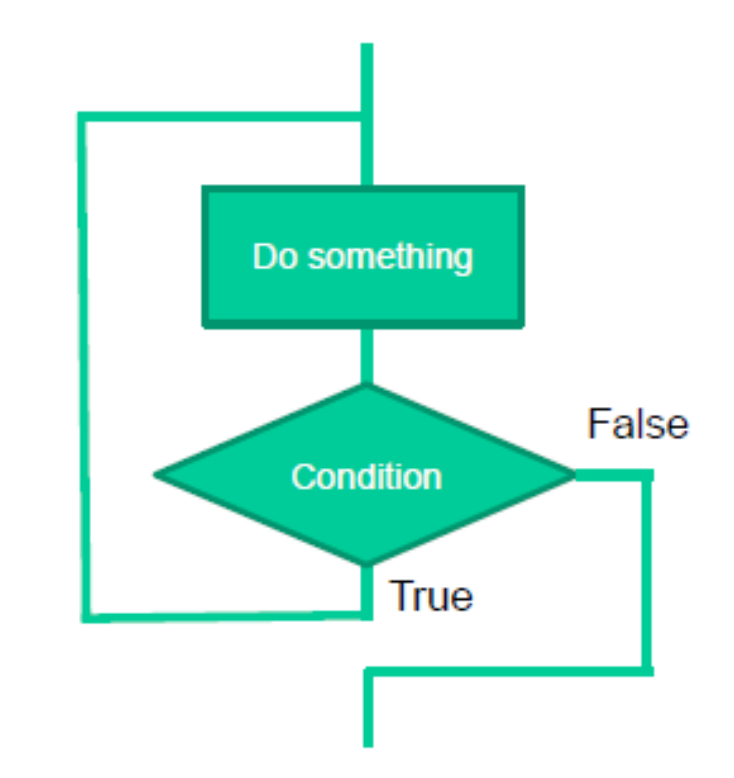
Pseudocode

Do

execute instruction(s)

WHILE condition

Flowchart:



DO-WHILE Loop Syntax

DO

    execute instruction(s)

WHILE(Condition)

```
int intCounter = 0; // initialize counter


do
{
printf( "%d ", intCounter ); // display counter
}
while ( ++ intCounter <= 10 ); //end do...while loop
```

Summary

Loop	Description
For loop	Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.
while loop	Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.
do…while loop	It is more like a while statement, except that it tests the condition at the end of the loop body.
nested loops	You can use one or more loops inside any other while, for, or do..while loop.

- Write a program that prints the following patterns separately
- Use for loops to generate the patterns