POLITEKNIK MALAYSIA
MUKAH

JTMK
JABATAN TEKNOLOGI MAKLUMAT DAN KOMUNIKASI

LOOPING STATEMENT IN PROGRAMMING

MC MICRO CREDENTIAL @ JTMK 2023

LOOPING IN PROGRAMMING

- Looping statement permits a sequence of the instructions to be executed repeatedly until certain condition is reached.

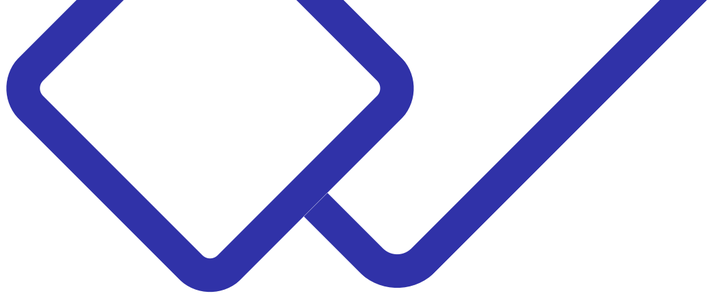
- Statements are controlled by a control loop or Boolean test.

- **Types** of looping statement:
  (a) while
  (b) do...while
  (c) for loop

- **Components** of looping:
  (a) initialization
  (b) termination/evaluation
  (c) increment/decrement

INITIALIZATION

- is the process of giving a variable an initial value.
- an initial value is important in the process of looping to count or sum the variable.
- initialization is the first value that will be evaluated.
- if the value is TRUE, the loop will repeat the statement.



LOOPING COMPONENTS

(a) initialization
(b) termination/evaluation
(c) increment/decrement

TERMINATION / EVALUATIONS

- used to verify how many loops will be done
- program starts by testing the Boolean test
- if the result is TRUE, the entire loop body is executed
- process will be repeated as long as the condition is TRUE
- if the result is FALSE, the loop body will not be executed

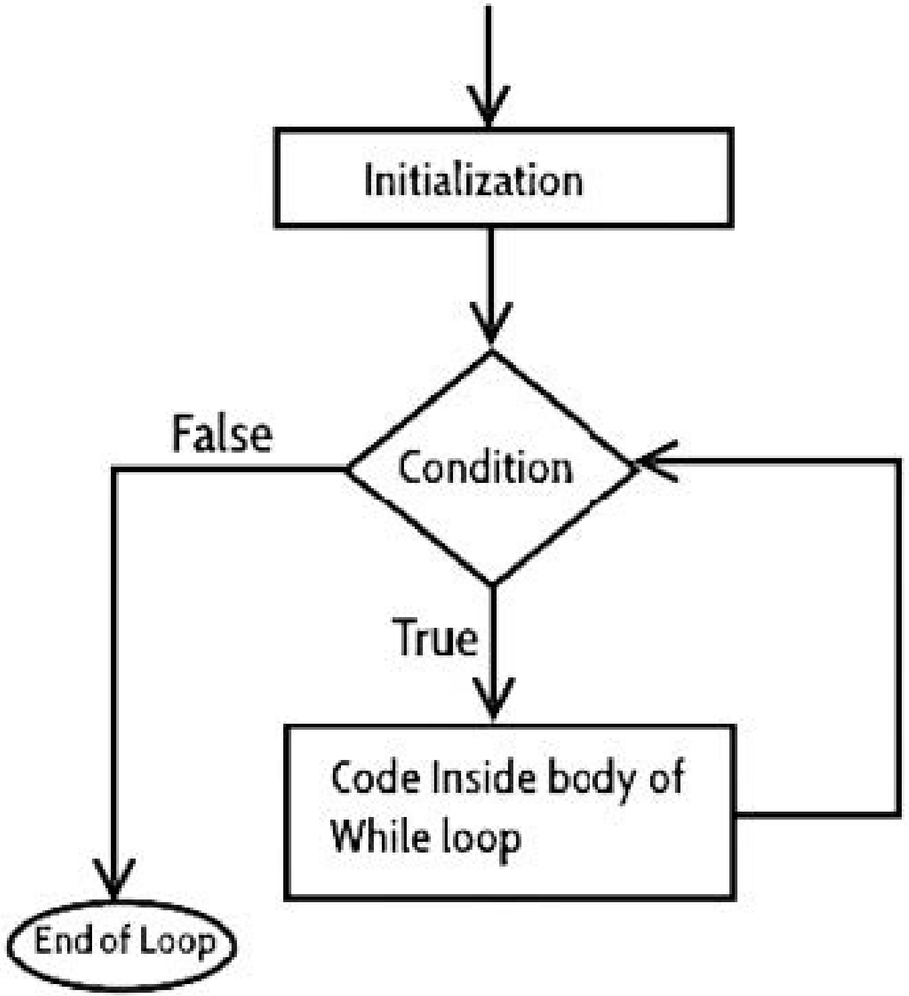
INCREMENT / DECREMENT

- used to update the initial value
- either increasing / decreasing value until the ending value is reached

TYPES OF LOOPING

WHILE STATEMENT

- pre-test loop
- condition is tested first
  - if value is TRUE: loop body will be repeated
  - if value is FALSE: loop will be skipped
- if the first evaluation produced FALSE value, the loop will no be executed
- the process repeated until condition is FALSE



SYNTAX

```
initial value;

while(condition/boolean test)
{ statements;
.........;
counter;
}
```

WHILE STATEMENT

Example of C++ and Java Programming

C++ PROGRAMMING

```cpp
//program to apply while statement

#include<iostream>
using namespace std;

int main(){
    int x=10;
    while(x>=0)
    {
        cout<<x<<endl;
        x=x-2;
    }
}
```

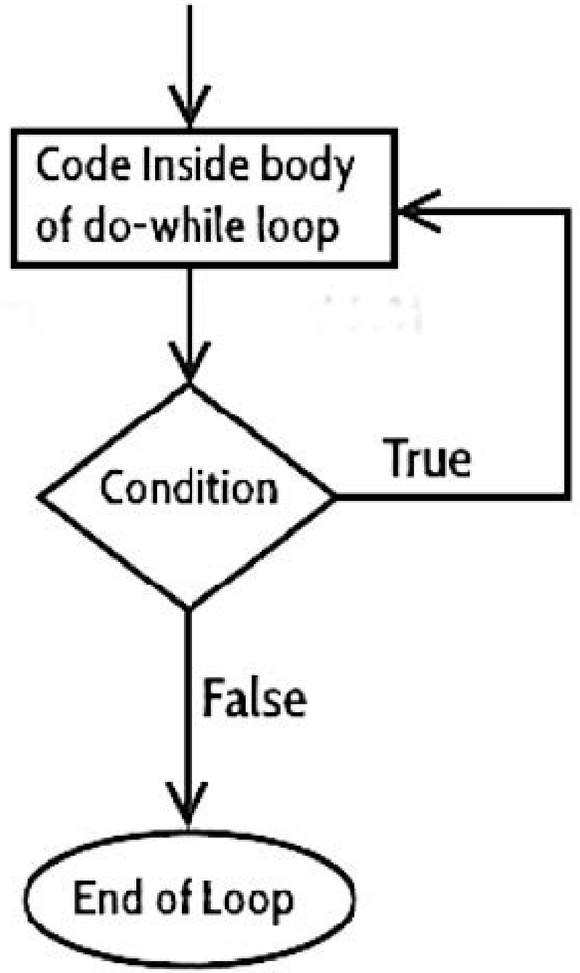
JAVA PROGRAMMING

```java
//to apply while loop
public class whileLoop {
    public static void main(String
        int x =10;
        while (x>=0){
            System.out.println(x);
            x=x-2;
        }
    }
}
```

TYPES OF LOOPING

DO...WHILE STATEMENT

- post-test loop
- loop body will be executed first
  condition will be tested later
    if value is TRUE: loop body will be repeated
    if value is FALSE: loop will be skipped
- if the first evaluation produced FALSE value, the loop will no be executed
- the process repeated until condition is FALSE
- loop is guaranteed to execute at least one time



SYNTAX

```
initial value;

do{
statements;
counter
}
while(condition/boolean test);
```

DO...WHILE STATEMENT

Example of C++ and Java Programming

C++ PROGRAMMING

```cpp
//program to apply do...while statement

#include<iostream>
using namespace std;

int main(){

    int count=0;
    do{
        cout<<count<<endl;
        count++;
    }
    while(count<=10);
}
```

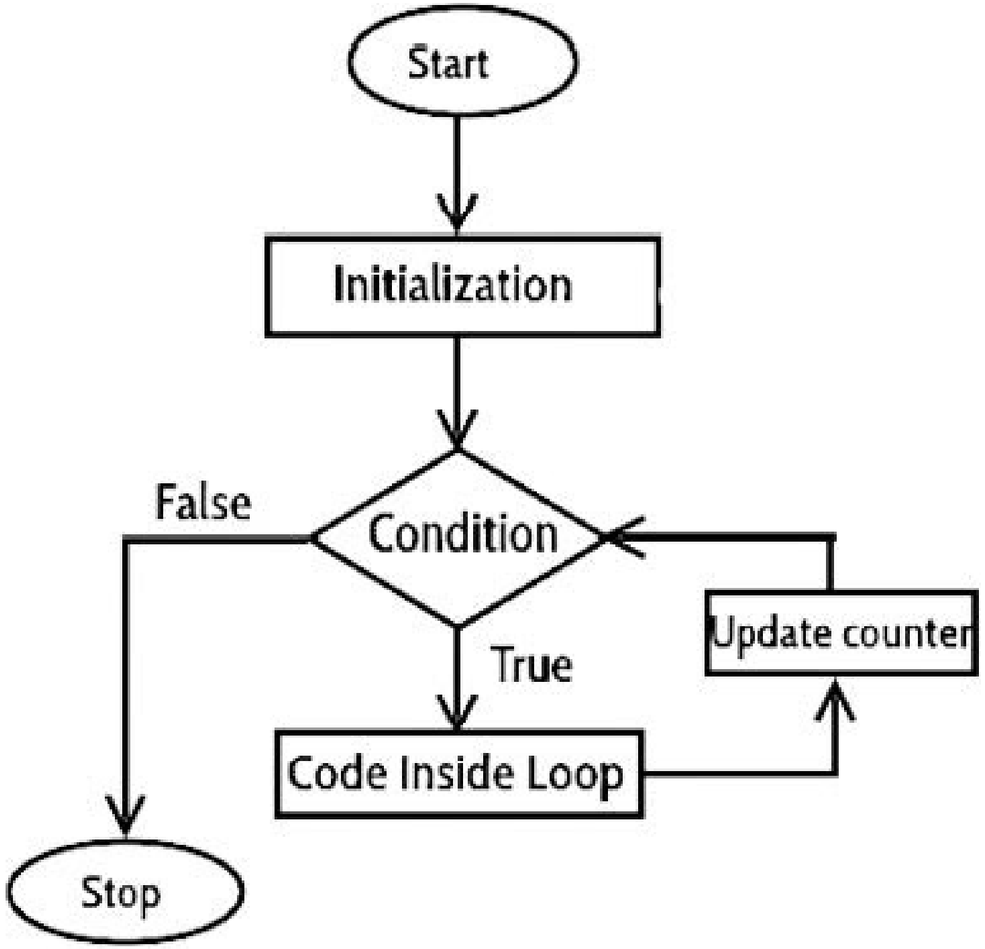
JAVA PROGRAMMING

```java
//to apply do while loop

public class doWhile {

    public static void main(String args[]){
        int count = 0;
        do {
            System.out.println(count);
            count++;
        } while (count <= 10);
    }
}
```

TYPES OF LOOPING

FOR LOOP STATEMENT

- pre-test loop
- condition is tested first
- if value is TRUE: loop body will be repeated
- if value is FALSE: loop will be skipped
- if the first evaluation produced FALSE value, the loop will no be executed
- the process repeated until condition is FALSE



SYNTAX

```
for(initial
value;condition;counter)

{
statements;
}
```

FOR LOOP STATEMENT

Example of C++ and Java Programming

C++ PROGRAMMING

```cpp
//program to apply for statement

#include<iostream>
using namespace std;

int main(){
    for(int x=25; x>=0; x=x-5)
    {
        cout<<" "<<x;
    }
}
```

JAVA PROGRAMMING

```java
//to apply for loop

public class forLoop {

    public static void main(String args[]){
        for (int x=25; x>0; x=x-5)
        {
            System.out.println(" " + x);
        }
    }
}
```