PROG0101
FUNDAMENTALS OF PROGRAMMING

Chapter 7
Loops

Topics

- Loops
- Condition Tested Loops
- Counted Loops
- Endless Loops
- FOR Loop
- WHILE Loop
- DO-WHILE Loop

Loops

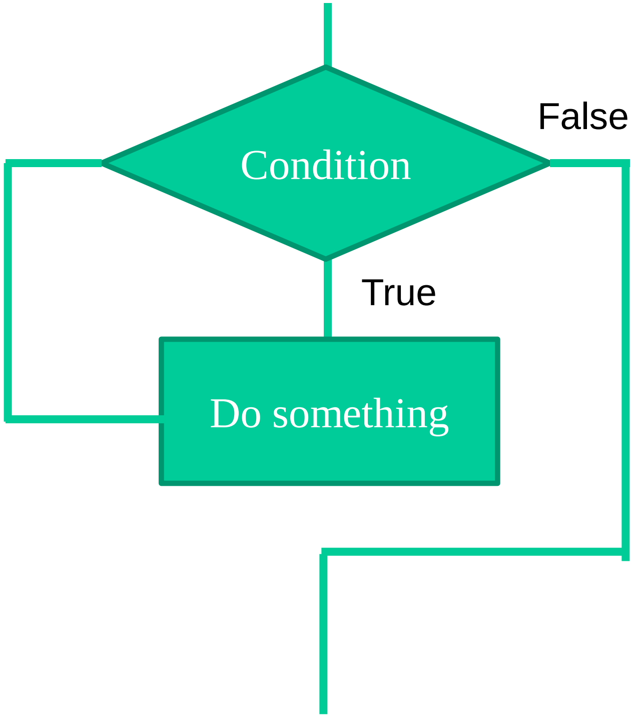
- A loop is a sequence of instructions that is continually repeated until a certain condition is reached.
- Loops allow for the same statement to be executed a number of times in succession.

Loops

Pseudocode:

```
Loop
    *Do Something*
Until *Condition*
```

Flowchart:



Loops

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

Condition Tested Loops

- A condition tested loop is one which repeats a set of instructions until a certain condition is reached.
- The test can be performed at the start of the loop (before any of the instructions are executed), during the loop, or at the end of the loop.
- Usually, the condition will be testing the result of executing the statements that are inside the loop.

Counted Loops

- A counted loop is one which allows the programmer to instruct the computer to perform a set of instructions *x* times, where *x* is usually an integer value, but some programming languages offer other data types.
- One could argue that the counted loop is just a condition tested loop which updates a counter and exits once a given value is reached.

Counted Loops

- The only time to use a count loop is when the program can determine ahead of time how many times the loop will repeat.
- 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, for example, where it is necessary to continually monitor for incoming messages from the operating system.

Example of Loop Statement

- These are examples 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

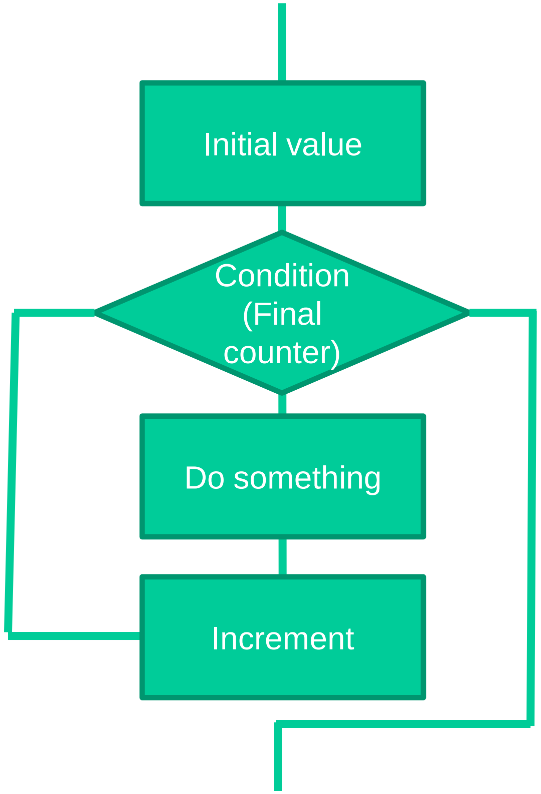
- The counter has the following three numeric values:
  - Initial counter value
  - Increment (the amount to add to the counter each time the loop runs)
  - Final counter value
- The loop ends when the counter reaches the final counter value, or, if there is an associated test condition, when the test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

FOR Loop

Pseudocode:

```
FOR x times
    Do Something
    Increment
```

Flowchart:



FOR Loop

FOR loop syntax:

```
FOR (initial counter value, final counter, increment)
    Statement (Do Something)
```

FOR Loop

Example 1:

```
FOR (x=1, x<5, x++)
    PRINT “Hello World”
```

```
Output:

Hello World
Hello World
Hello World
Hello World
```

FOR Loop

Example 2:

```
FOR (x=1, x<=4, x++)
    PRINT x
```

Output:

```
1
2
3
4
```

FOR Loop

Example 3:

```
FOR (x=5, x>0, x--)
      PRINT x
```

```
Output:

5
4
3
2
1
```

WHILE Loop

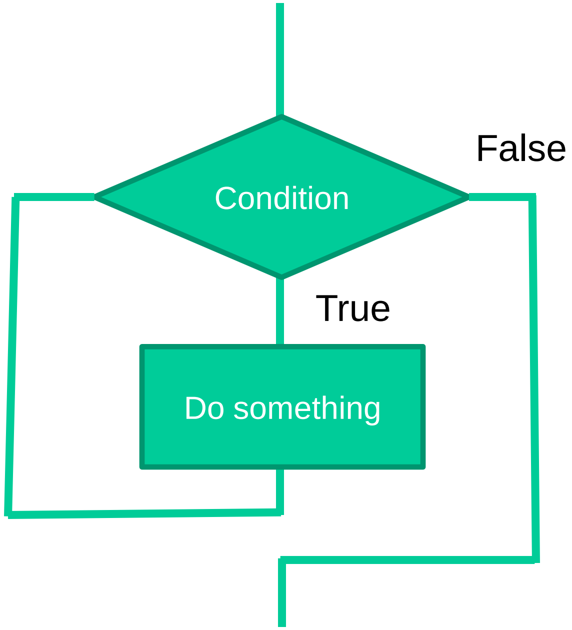
- A WHILE loop is a loop that repeats while some condition is satisfied.
- The WHILE loop tests its condition at the beginning of every loop.
- If the condition is false from the start, the sequence of activities contained in the loop never runs at all.

WHILE Loop

Pseudocode:

```
WHILE condition
    Do Something
```

Flowchart:



WHILE Loop

WHILE loop syntax:

```
WHILE (Condition)
    Statement (Do Something)
```

WHILE Loop

Example 1:

```
WHILE (x < 5)
    PRINT “Hello World”
    x++
```

Output:

```
Hello World
Hello World
Hello World
Hello World
```

WHILE Loop

Example 2:

```
WHILE (key != Esc)
    PRINT “Hello World”
```

Output:

```
Hello world
Hello world
Hello world
Hello world
…
```

DO-WHILE Loop

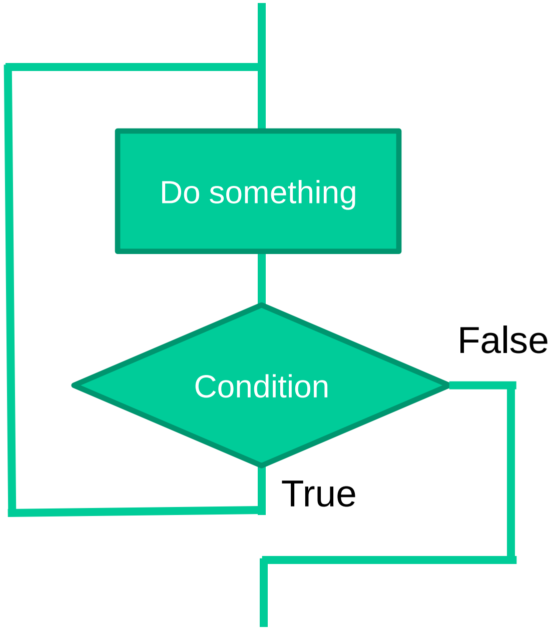
- Like a while loop, a do-while loop is a loop that repeats while some condition is satisfied.
- Unlike a while loop, a do-while loop tests its condition at the end of the loop.
- This means that its sequence of activities always runs at least once.

DO-WHILE Loop

Pseudocode:

```
Do
    something
WHILE condition
```

Flowchart:



DO-WHILE Loop

DO-WHILE Loop Syntax

```
DO
     Statement
WHILE (Condition)
```

Loops

DO-WHILE Loop

Example 1:

```
x=1
DO
    PRINT “Hello World”
    x++
WHILE (x<5)
```

Output:

```
Hello World
Hello World
Hello World
Hello World
```

DO-WHILE Loop

Example 2:

```
x=1
DO
    PRINT “Hello World”
    x++
WHILE (x>5)
```

Output:

```
Hello World
```

DO-WHILE Loop

Example 3:

```
DO
    PRINT “Hello World”
WHILE (Key != Esc)
```

Output:

```
Hello World
Hello World
Hello World
…
```

Exercise

Draw flowchart diagram for the following programs using loop:

1. A program that display number 1 to 20
2. A program that display a person name x times.