UNIVERSITY of SAN CARLOS
SCIENTIA • VIRTUS • DEVOTIO

DCISM
DEPARTMENT OF COMPUTER, INFORMATION SCIENCES AND MATHEMATICS

CIS 1101 – PROGRAMMING 1

CONTROL STRUCTURES

Part 2

LOOP STRUCTURE IN C: ITERATION

- It is where a **set of instructions or structures are repeated** in a sequence a specified number of times or until a condition is met.
- It is also known as **iteration**.

- Used in programming to execute a block of code repeatedly until a specified condition is met.

LOOP STRUCTURE IN C: PRETEST LOOPS

condition is checked before each repetition to determine if the loop should terminate or continue

FOR LOOP: DEFINITION

CONTROL STRUCTURES IN C

- Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.

FOR LOOP: SYNTAX

CONTROL STRUCTURES IN C

```
for (initialization; test condition; increment/decrement)
{
    /*  statements inside the body of loop */
}
```

FOR LOOP: HOW DOES IT WORK?

- The initialization statement is executed only once.
- Then, the test expression (condition) is evaluated.
  - If the test expression (condition) is evaluated to false, the for loop is terminated.
  - If the test expression (condition) is evaluated to true, statements inside the body of for loop are executed, and the update expression is updated.

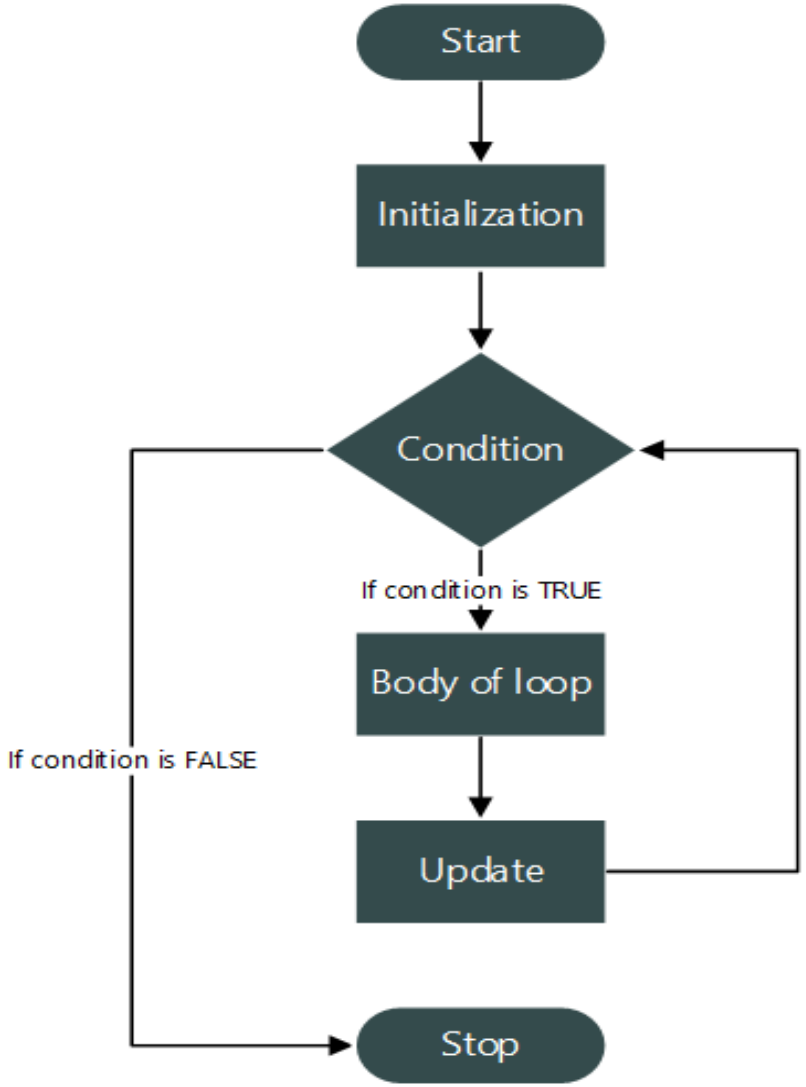
FOR LOOP: HOW DOES IT WORK?

CONTROL STRUCTURES IN C

- Then, again the test expression (condition) is evaluated.
- The process goes on until the test expression (condition) is false.
  - When the test expression is false, the loop terminates.

FOR LOOP: FLOWCHART

CONTROL STRUCTURES IN C



FOR LOOP: EXAMPLE

CONTROL STRUCTURES IN C

```
#include <stdio.h>

int main ()
{
   /* local variable definition */
   int y;
   /* for loop execution */
   for(y=1; y<15; y++ )
   {
      printf("The value of y:  %d\n", y);
   }

    return 0;
}
```

INFINITE LOOP

- It happens when a condition never becomes false.
- The for loop is traditionally used for this purpose.
- Since none of the three expressions that form the 'for' loop are required, you can make an endless loop by leaving the conditional expression empty.

```
#include <stdio.h>

/* NOTE - You can terminate an infinite loop */
/* by pressing Ctrl + C keys or Ctrl + Break keys.*/

int main ()
{

    for( ; ; )
    {
      printf("This loop will run forever.\n");
     }

    return 0;
}
```

WHILE: DEFINITION

CONTROL STRUCTURES IN C

- Repeats a statement or group of statements while a given condition is true.

- Tests the condition before executing the loop body.

WHILE: SYNTAX

CONTROL STRUCTURES IN C

```
while (test condition)
{
    /* statements inside the body of the loop */
}
```

WHILE: HOW DOES IT WORK?

- The **while loop** evaluates the test expression inside the parenthesis ().

- If the test expression is true, statements inside the body of while loop are executed.

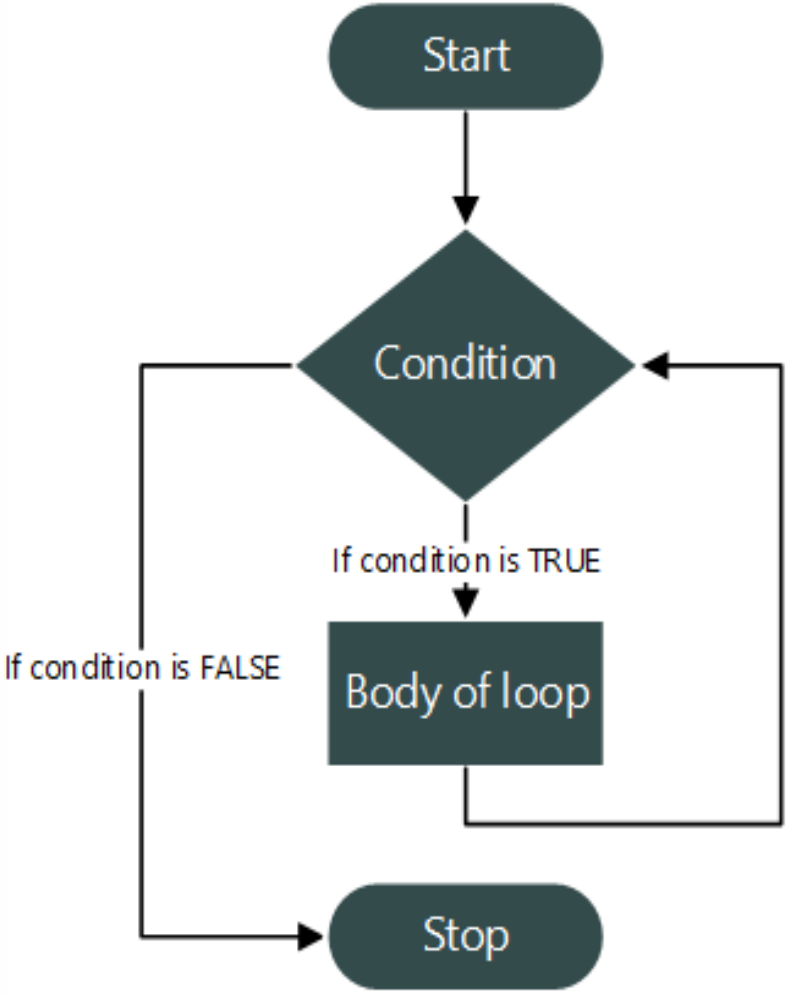
WHILE: HOW DOES IT WORK?

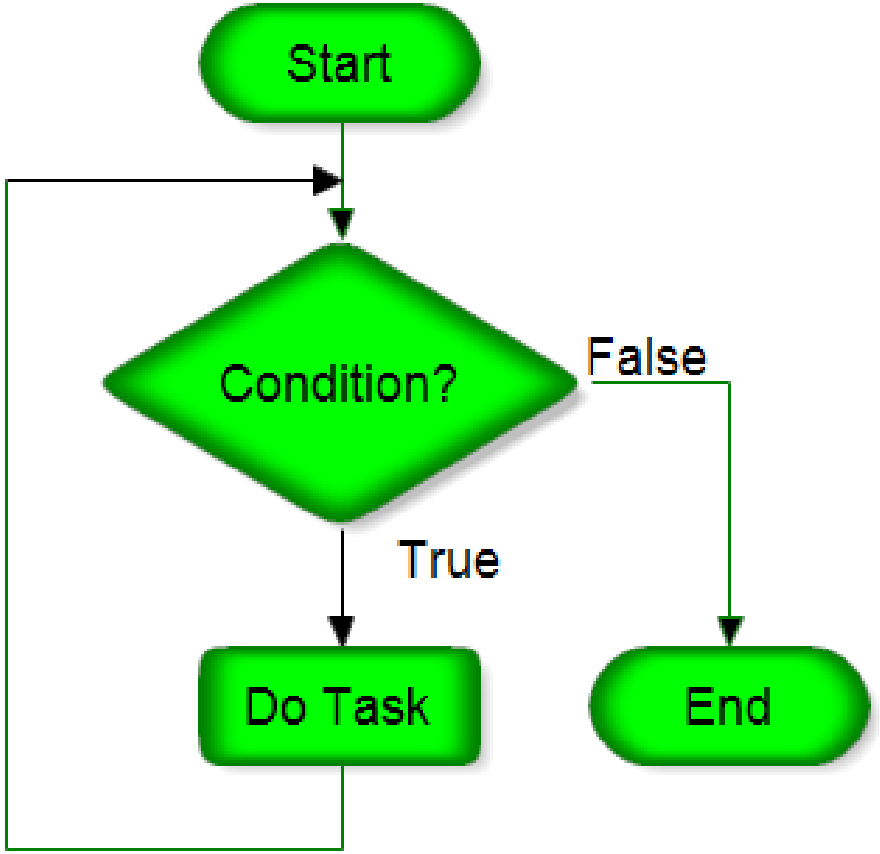
CONTROL STRUCTURES IN C

- Then, the test expression is evaluated again.
- The process goes on until the test expression is evaluated to false.
- If the test expression is false, the loop terminates.

WHILE: FLOWCHART

CONTROL STRUCTURES IN C





WHILE: EXAMPLE

CONTROL STRUCTURES IN C

```c
#include <stdio.h>

int main ()
{

   /* local variable definition */
   int y = 1;
```

```c
/* while loop execution */
   while( y < 15 )
   {
     printf("The value of y:  %d\n", y);
     y++;
    }

    return 0;
}
```

LOOP STRUCTURE IN C: POSTTEST LOOPS

condition is checked after each repetition to determine if loop should terminate or continue

DO-WHILE: DEFINITION

CONTROL STRUCTURES IN C

- The **do…while loop** is similar to while loop with one important difference.

- The body of the **do…while loop** is executed at least once.
- Only then, the test expression is evaluated.

DO-WHILE: SYNTAX

CONTROL STRUCTURES IN C

```
do
{
    /* statements inside the body of the loop */
}while (test condition);
```

DO-WHILE: HOW DOES IT WORK?

CONTROL STRUCTURES IN C

- The body of **do...while loop** is executed at least once and only then, the test expression is evaluated.

- If the test expression is true, the body of the loop is executed again and the test expression is evaluated.

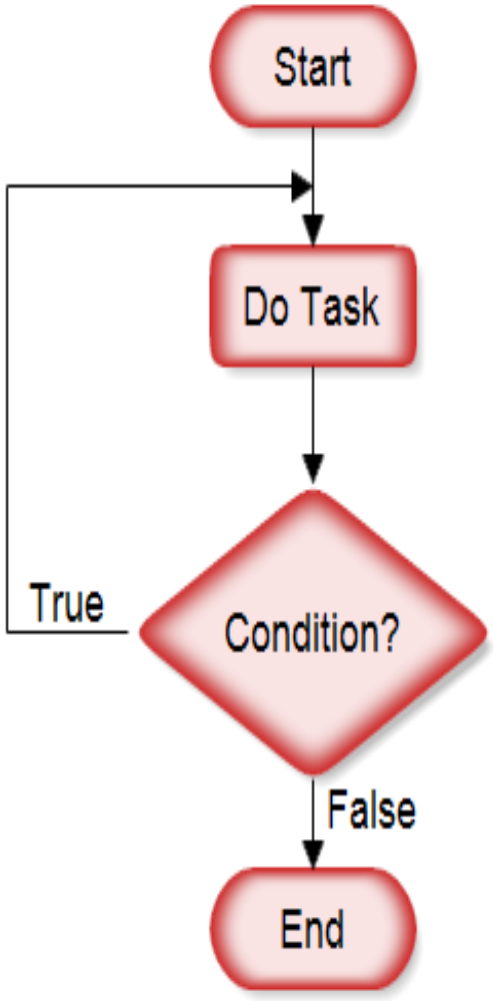
DO-WHILE: HOW DOES IT WORK?

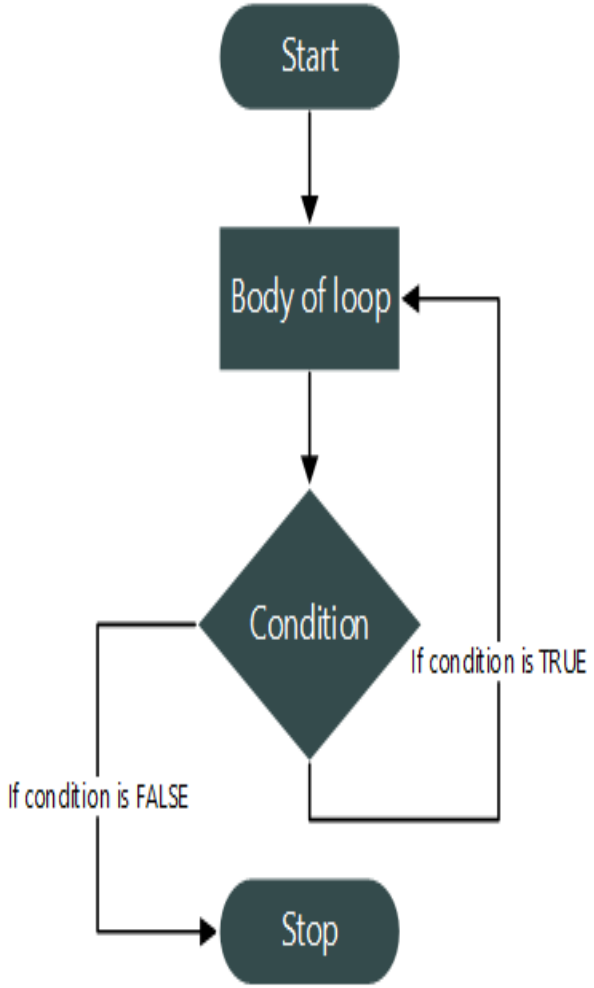
CONTROL STRUCTURES IN C

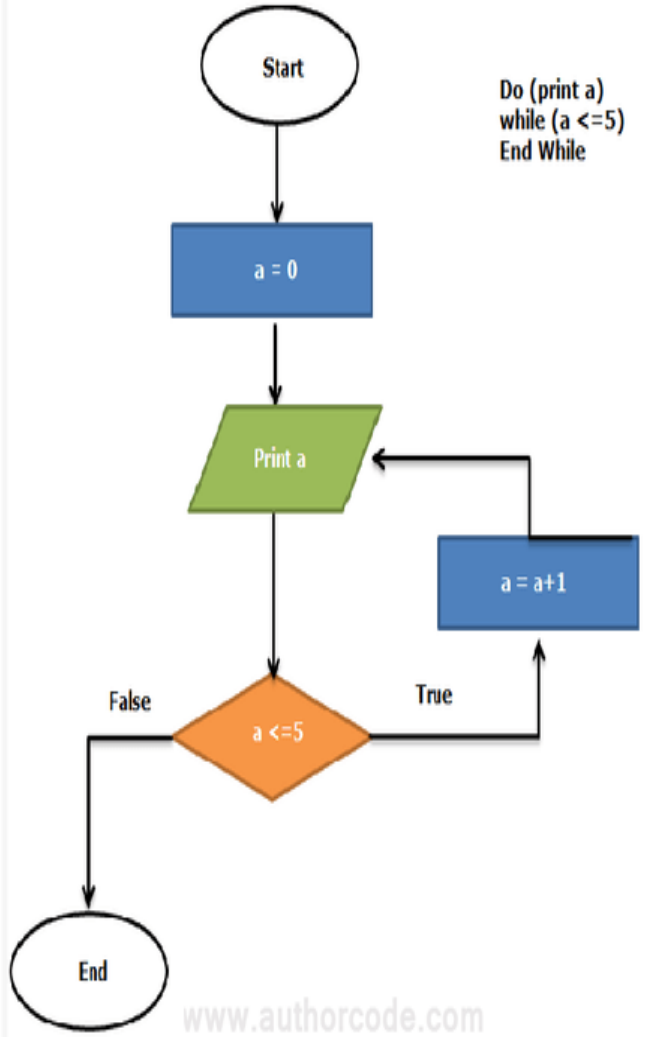
- This process goes on until the test expression becomes false.
- If the test expression is false, the loop ends.

DO-WHILE: FLOWCHART

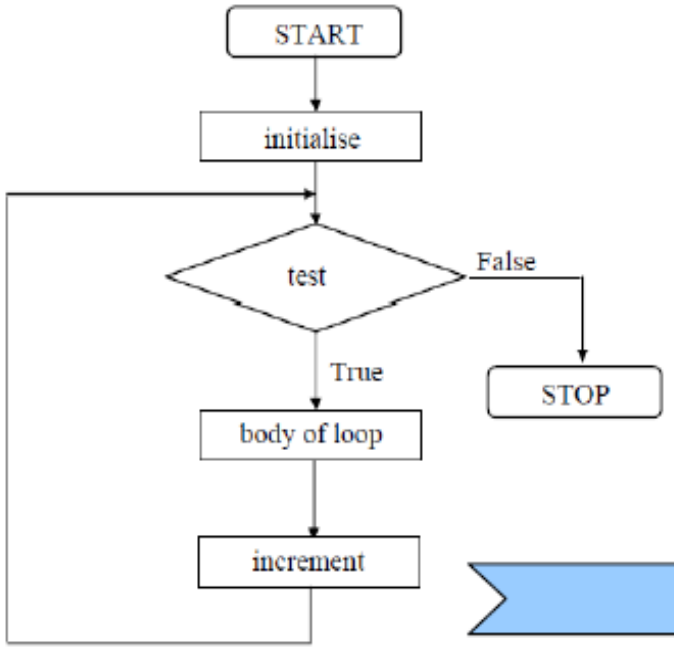
CONTROL STRUCTURES IN C

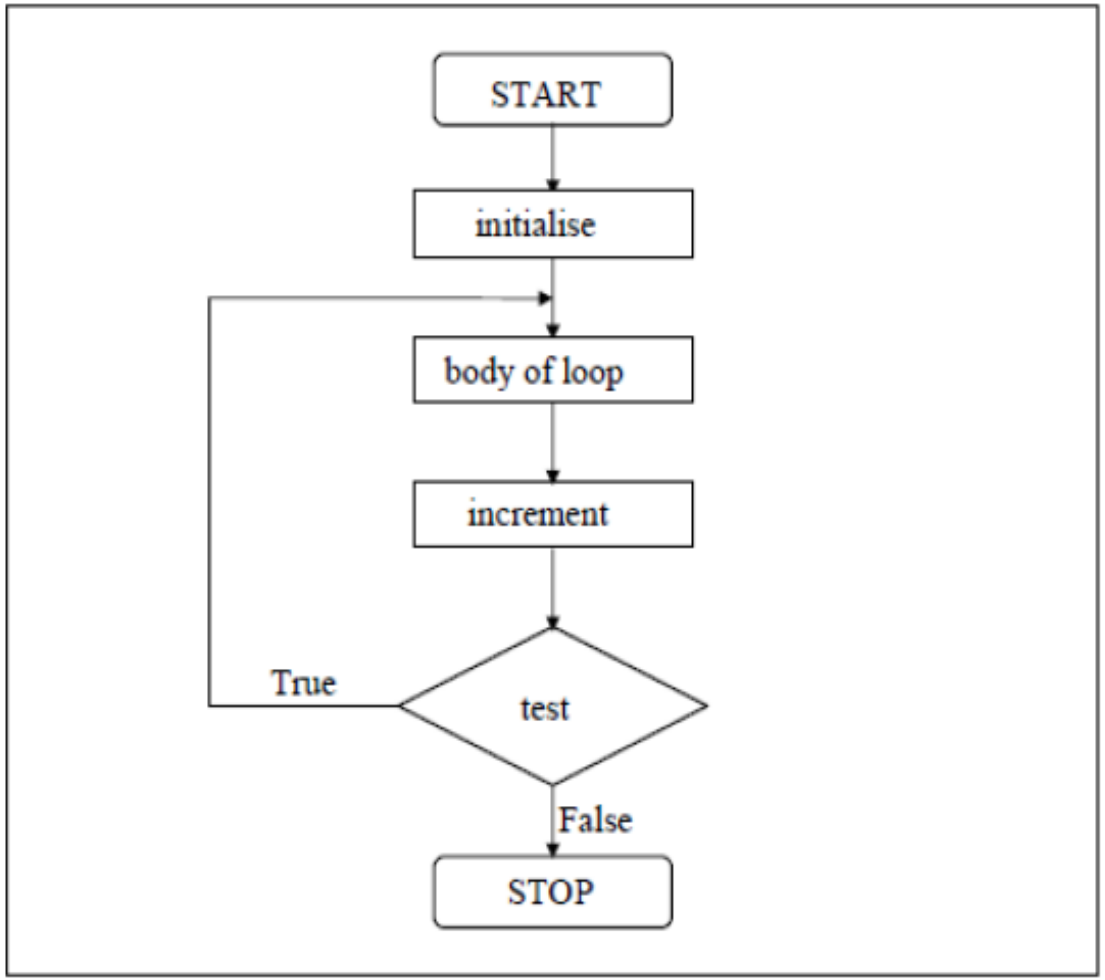






WHILE VERSUS DO-WHILE





CONTROL STRUCTURES IN C

DO-WHILE: EXAMPLE

CONTROL STRUCTURES IN C

```
#include <stdio.h>

int main ()
{

   /* local variable definition */
   int y = 1;

   /* do loop execution */
   do
   {
       printf("The value of y:  %d\n", y);
       y = y + 1;
   }while( y < 15 );

   return 0;
}
```