Topic 4 : Decision Making, Branching and Looping

[Q] Explain IF statement with syntax, program and flowchart.

Ans:

C if statement

```
if (testExpression)
{
   // statements
}
```

The **if** statement evaluates the test expression inside parenthesis.

If test expression is evaluated to true (nonzero), statements inside the body of **if** is executed.

If test expression is evaluated to false (0), statements inside the body of **if** is skipped.

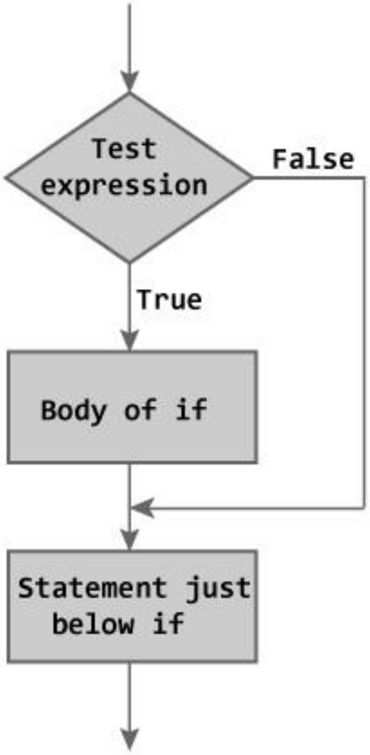


Figure: Flowchart of if Statement

```
// Program to display a number if user enters negative number
// If user enters positive number, that number won't be displayed

#include <stdio.h>
int main()
{
    int number;

    printf("Enter an integer: ");
    scanf("%d", &number);

    // Test expression is true if number is less than 0
    if (number < 0)
    {
        printf("You entered %d.\n", number);
    }

    printf("The if statement is easy.");

    return 0;
}
```

Output 1

```
Enter an integer: -2
You entered -2.
The if statement is easy.
```

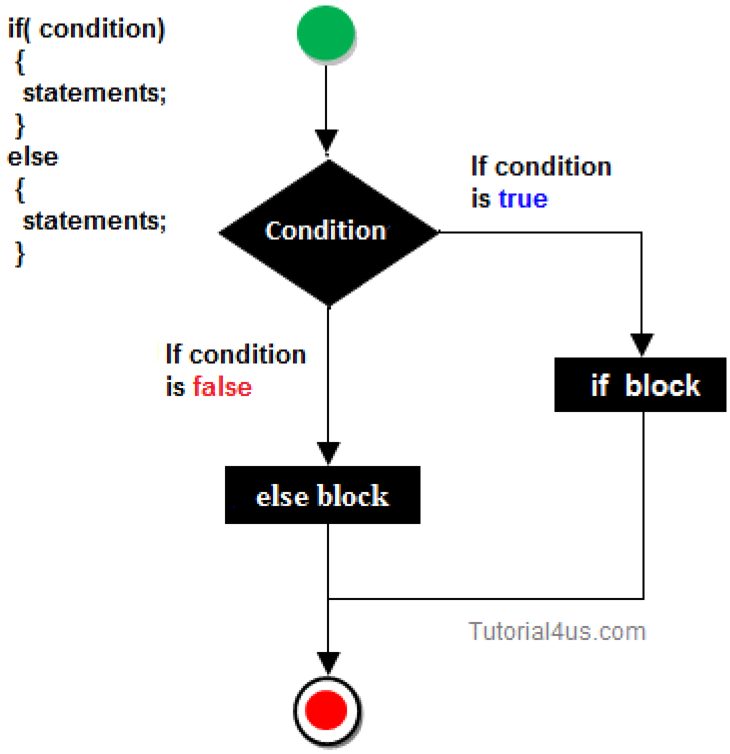
When user enters -2, the test expression (number < 0) becomes true. Hence, You entered -2 is displayed on the screen.

Output 2

```
Enter an integer: 5
The if statement in C programming is easy.
```

[Q] Explain IF Else statement with syntax, program and flowchart.

Ans: In general it can be used to execute one block of statement among two blocks, in C language if and else are the keyword in C.



```
if( condition)
 {
  statements;
 }
else
 {
  statements;
 }
```

C if...else statement

```
// Program to check whether an integer entered by the user is odd or even

#include <stdio.h>
int main()
{
    int number;
    printf("Enter an integer: ");
    scanf("%d",&number);

    // True if remainder is 0
    if( number%2 == 0 )
        printf("%d is an even integer.",number);
    else
        printf("%d is an odd integer.",number);
    return 0;
}
```

Output

```
Enter an integer: 7
7 is an odd integer.
```

When user enters 7, the test expression **(number%2 == 0)** is evaluated to false. Hence, the statement inside the body of **else** statement **printf("%d is an odd integer");** is executed and the statement inside the body of **if** is skipped.

[Q] Write the syntax of nested if else statement.

Ans: Nested if else statement:

When a series of decisions are involved, we may have to use more than one if…else statement in nested form as shown below:

Syntax:

```
if(test condition1)
{
if(test condition2)
{
statement-1;
}
else
{
statement-2;
}
}
else
{
statement-3;
}
statement-x;
```

The nested if...else statement allows you to check for multiple test expressions and execute different codes for more than two conditions.

```c
// Program to relate two integers using =, > or <
#include <stdio.h>
int main()
{
    int number1, number2;
    printf("Enter two integers: ");
    scanf("%d %d", &number1, &number2);

    //checks if two integers are equal.
    if(number1 == number2)
    {
        printf("Result: %d = %d",number1,number2);
    }
    //checks if number1 is greater than number2.
    else if (number1 > number2)
    {
        printf("Result: %d > %d", number1, number2);
    }

    // if both test expression is false
    else
    {
        printf("Result: %d < %d",number1, number2);
    }
    return 0;
}
```

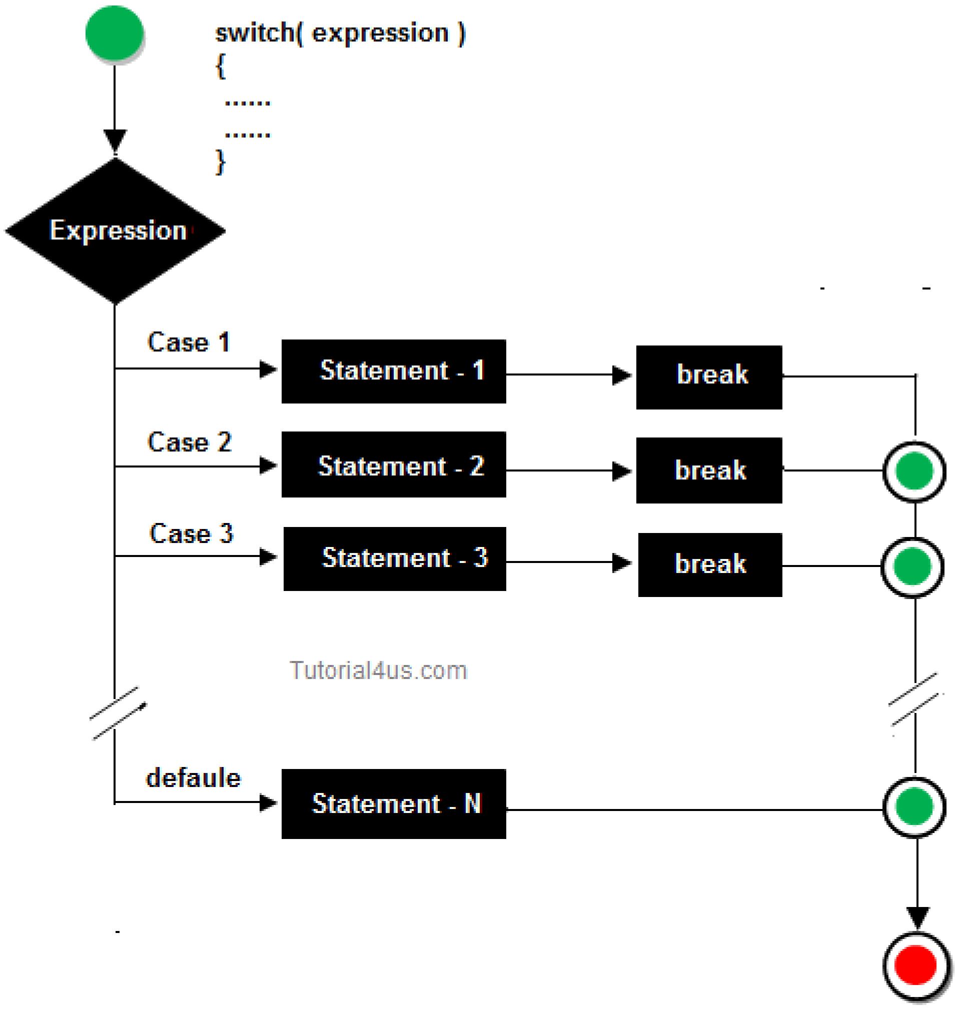
Output

```
Enter two integers: 12

23

Result: 12 < 23
```

4.5 Else IF ladder

[Q] Explain Switch Statement with flowchart and syntax and program.

Ans: **A switch statement work with byte, short, char and int primitive data type, it also works with enumerated types and string.**



Syntax :

```
switch(expression/variable)

{

case  value:

//statements

// any number of case statements

break;  //optional

default: //optional

//statements

}
```

Rules for apply switch

With switch statement use only byte, short, int, char data type.

We can use any number of case statements within a switch.

Value for a case must be same as the variable in switch .

Limitations of switch

Logical operators cannot be used with switch statement.

Switch case variables can have only int and char data type. So float data type is not allowed.

Program

```
#include<stdio.h>

#include<conio.h>
```

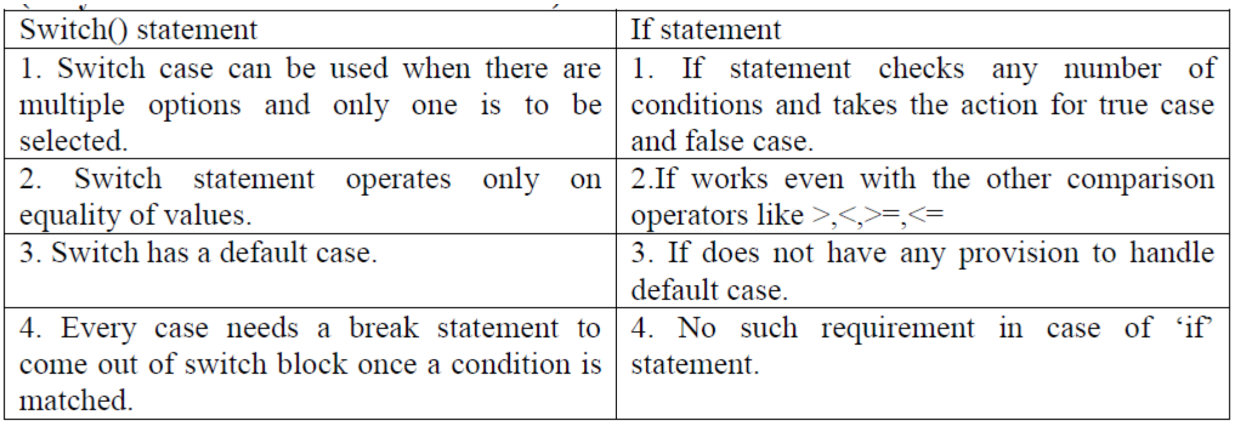
```
void main()

{

int ch;

clrscr();

printf("Enter any number (1 to 7)");

scanf("%d",&ch);

switch(ch)

{

case  1:

printf("Today is Monday");

break;

case  2:

printf("Today is Tuesday");

break;

case  3:

printf("Today is Wednesday");

break;

case  4:

printf("Today is Thursday");

break;

case  5:

printf("Today is Friday");

break;
case  6:

printf("Today is Saturday");

break;

case  7:

printf("Today is Sunday");

break;

default:

printf("Only enter value 1 to 7");

}

getch();

}
```

Output :

```
Enter any number (1 to 7): 5

Today is Friday
```

[Q] State two difference between switch statement and if statement.

Ans:



Switch() statement	If statement
1. Switch case can be used when there are multiple options and only one is to be selected.	1. If statement checks any number of conditions and takes the action for true case and false case.
2. Switch statement operates only on equality of values.	2.If works even with the other comparison operators like >,<,>=,<=
3. Switch has a default case.	3. If does not have any provision to handle default case.
4. Every case needs a break statement to come out of switch block once a condition is matched.	4. No such requirement in case of 'if' statement.

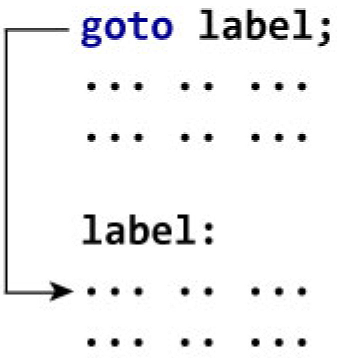
[Q] Explain GOTO statement with syntax and program .

Ans:

Syntax of goto statement

```
goto label;
... .. ...
... .. ...
... .. ...
label:
statement;
```

The label is an identifier. When goto statement is encountered, control of the program jumps to label: and starts executing the code.



```
goto label;
... .. ...
... .. ...

label:
... .. ...
... .. ...
```

```
// Program to calculate the sum and average of maximum of 5 numbers
// If user enters negative number, the sum and average of previously
entered positive number is displayed

# include <stdio.h>

int main()
{

    const int maxInput = 5;
    int i;
    double number, average, sum=0.0;
    for(i=1; i<=maxInput; ++i)
    {
        printf("%d. Enter a number: ", i);
        scanf("%lf",&number);

    // If user enters negative number, flow of program moves to label jump
        if(number < 0.0)
            goto jump;

        sum += number; // sum = sum+number;
    }

    jump:

    average=sum/(i-1);
    printf("Sum = %.2f\n", sum);
    printf("Average = %.2f", average);

    return 0;
}
```

Output

```
1. Enter a number: 3
2. Enter a number: 4.3
3. Enter a number: 9.3
4. Enter a number: -2.9
Sum = 16.60
```

[Q] Why use loop ?

Ans: When you need to execute a block of code several number of times then you need to use looping concept in C language.

Advantage with looping statement

Reduce length of Code

Take less memory space.

Burden on the developer is reducing.

Time consuming process to execute the program is reduced.

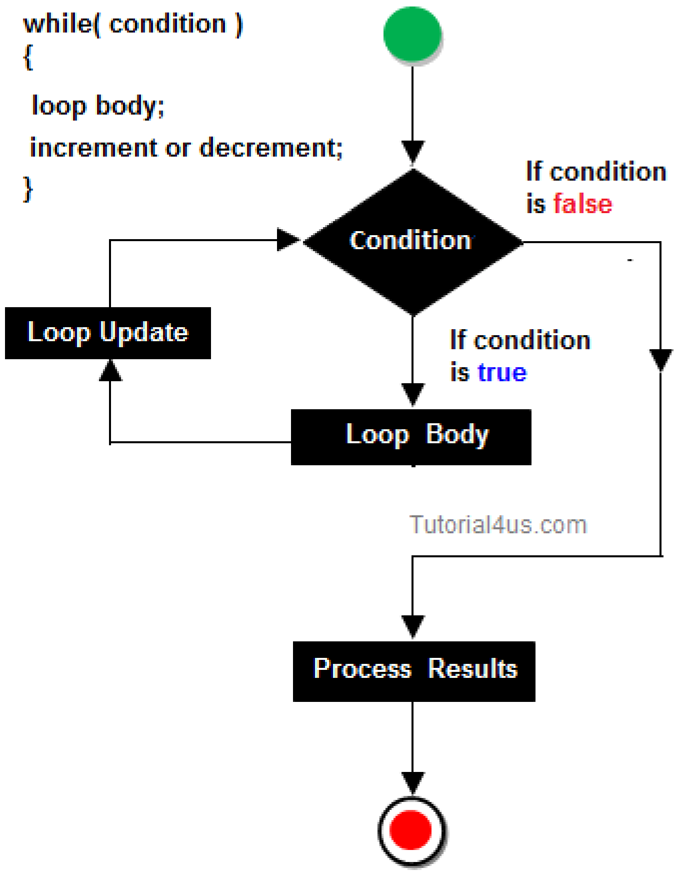
Types of Loops.

There are three type of Loops available in 'C' programming language.

while loop
for loop
do..while

[Q] Explain while Loop with syntax ,flowchart and program.

Ans: While loop
In while loop First check the condition if condition is true then control goes inside the body otherwise goes outside the body. while loop will be repeats in clock wise direction.



```
while( condition )
{
 loop body;
 increment or decrement;
}
```

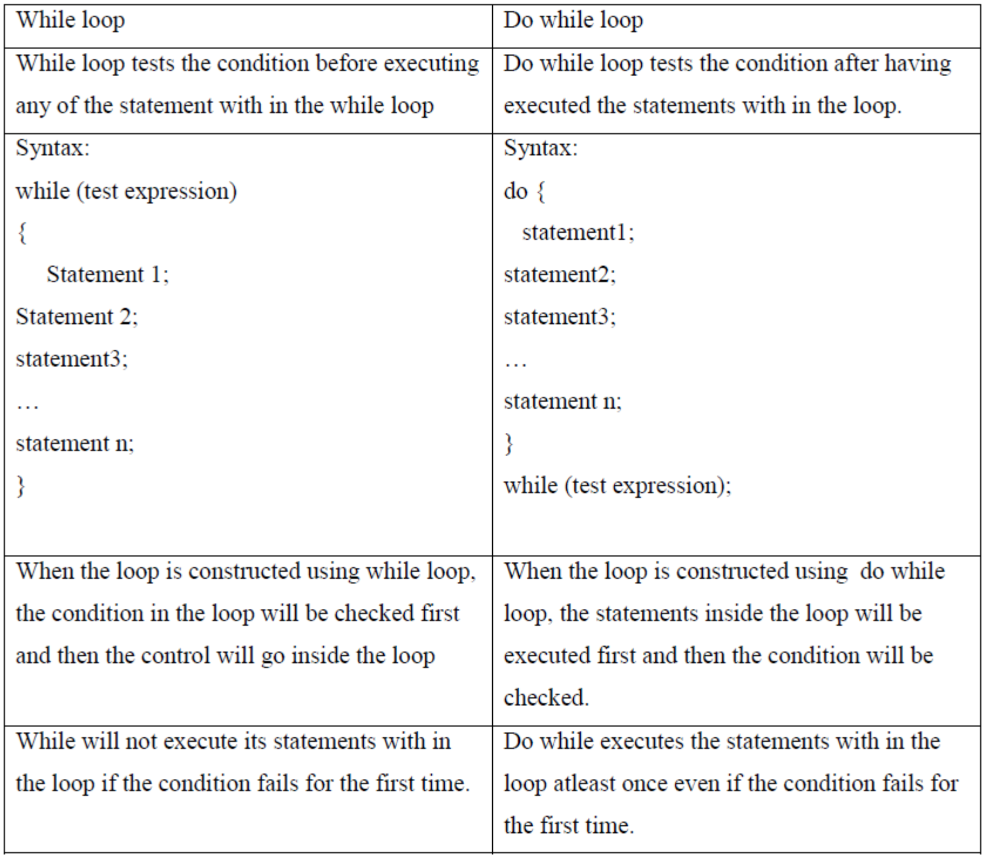
Syntax ;
Assignment;
while(condition)
{
Statements;
......
Increment/decrements (++ or --);

```
}
#include<stdio.h>
#include<conio.h>

void main()
{
int i;
clrscr();
i=1;
while(i<5)
{
printf("\n%d",i);
i++;
}
getch();
}
```

Output

```
1
2
3
4
```

[Q] Distinguish between while and do-while loop.



While loop	Do while loop
While loop tests the condition before executing any of the statement with in the while loop	Do while loop tests the condition after having executed the statements with in the loop.

Syntax:
while (test expression)
{
Statement 1;
Statement 2;
statement3;
…
statement n;
}
	Syntax:
	do {
	statement1;
	statement2;
	statement3;
	…
	statement n;
	}
	while (test expression);

When the loop is constructed using while loop, the condition in the loop will be checked first and then the control will go inside the loop	When the loop is constructed using do while loop, the statements inside the loop will be executed first and then the condition will be checked.
While will not execute its statements with in the loop if the condition fails for the first time.	Do while executes the statements with in the loop atleast once even if the condition fails for the first time.

[Q] What is looping? List various looping constructs of C.

Ans:

Looping:

C executes instructions or statements sequentially one after other and one instruction is executed only
once. Sometimes it is required to execute same instructions again and again which are already
executed. Such situation is called as looping.

various looping constructs of C:-

1. for loop
2.while loop

[Q] State do while loop with example.

Ans: DO..WHILE loops are useful for things that want to loop at least once.

Syntax:

```
do
{
/* body of do while */
} (condition);
```

The condition is tested at the end of the block instead of the beginning, so the block will be executed at least once.

If the condition is true, we jump back to the beginning of the block and execute it again.

A do..while loop is almost the same as a while loop except that the loop body is guaranteed to execute at least once.

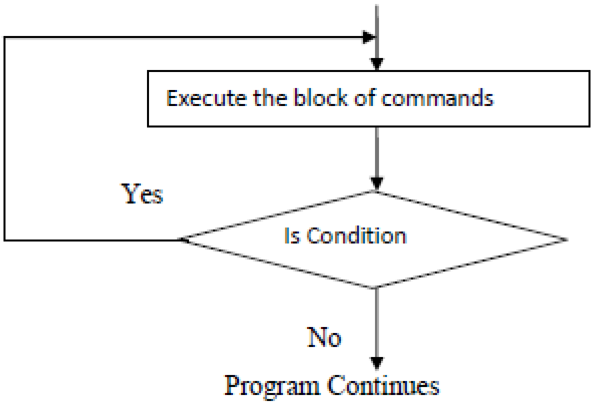
A while loop says "Loop while the condition is true, and execute this block of code", a do..while loop says "Execute this block of code, and then continue to loop while the condition is true

Example:

```
main( )
{
int i = 0;
do
{
printf("HELLO");
}While (i!=0);
}
```

OUTPUT: HELLO

Even though the condition is false the Hello is printed once.



[Q] Explain DO WHILE Loop with syntax ,flowchart and program

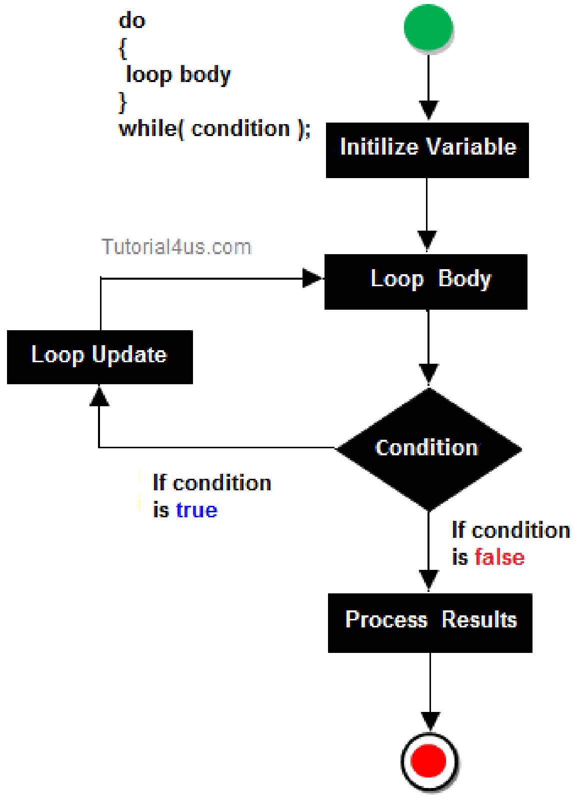
Ans : do-while

A do-while loop is similar to a while loop, except that a do-while loop is execute at least one time.

A do while loop is a control flow statement that executes a block of code at least once, and then repeatedly executes the block, or not, depending on a given condition at the end of the block (in while).

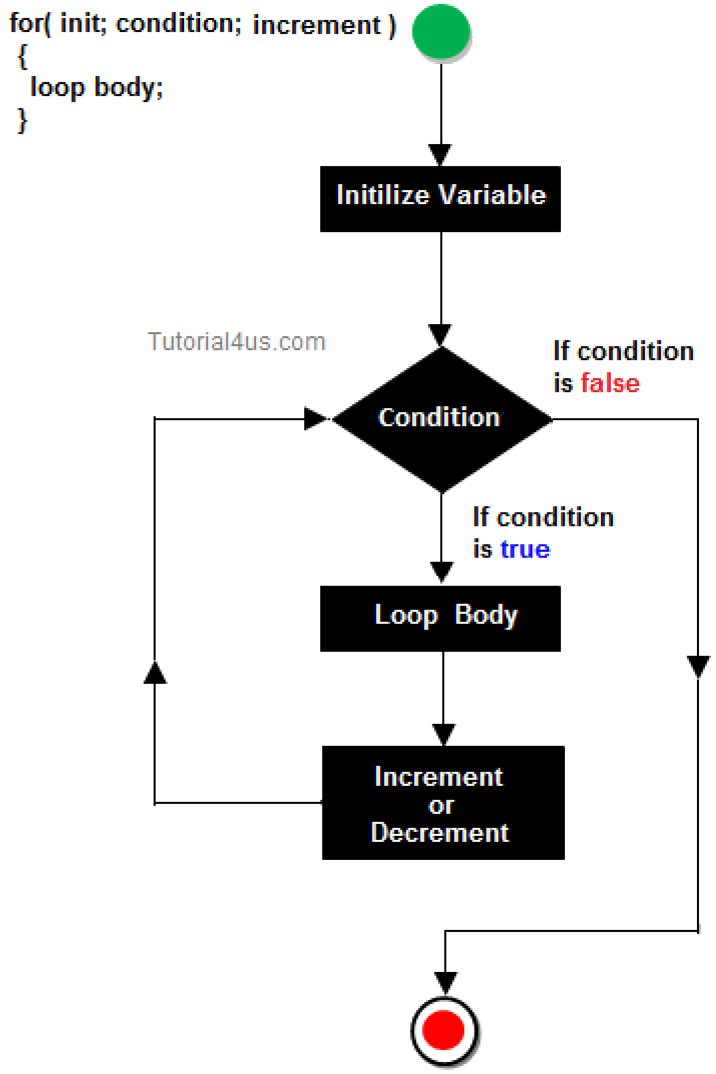
Syntax :

```
do
{
Statements;
........
Increment/decrement (++ or --)
} while();
```



[Q] Explain FOR Loop with syntax ,flowchart and program
Ans: For loop

for loop is a statement which allows code to be repeatedly executed. For loop contains 3 parts Initialization, Condition and Increment or Decrements.



```
for( init; condition; increment )
 {
   loop body;
 }
```

```
#include<stdio.h>
#include<conio.h>

void main()
{
int i;
clrscr();
for(i=1;i<5;i++)
{
```

```
printf("\n%d",i);
}
getch();
}
```

Output

```
1
2
3
4
```

=================AND THAT IS END OF 4 CHAPTER ====