SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE NAME : 23CST101 C PROGRAMMING AND DATA STRUCTURES

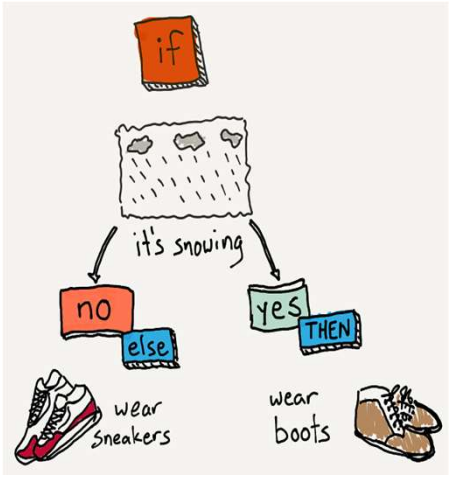
I YEAR /II SEMESTER

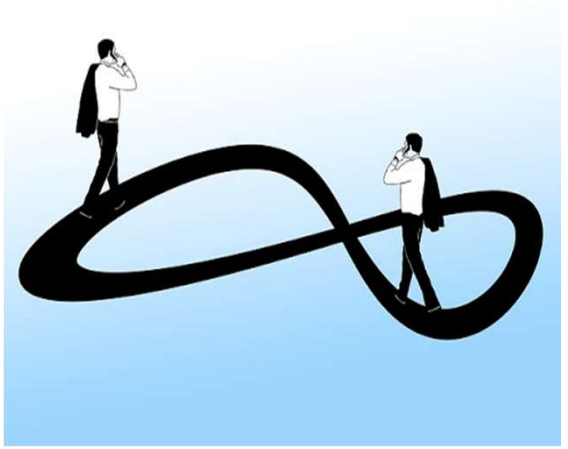
Unit 1- C PROGRAMMING FUNDAMENTALS- A REVIEW

Topic 7: Looping statements

Brain Storming

1. How Decision making and Iterative statements are executed in C?





Loops in C

•In any programming language including C, loops are used to execute a set of statements repeatedly until a particular condition is satisfied.

•**Types of Loop**

•There are 3 types of Loop in C language, namely:

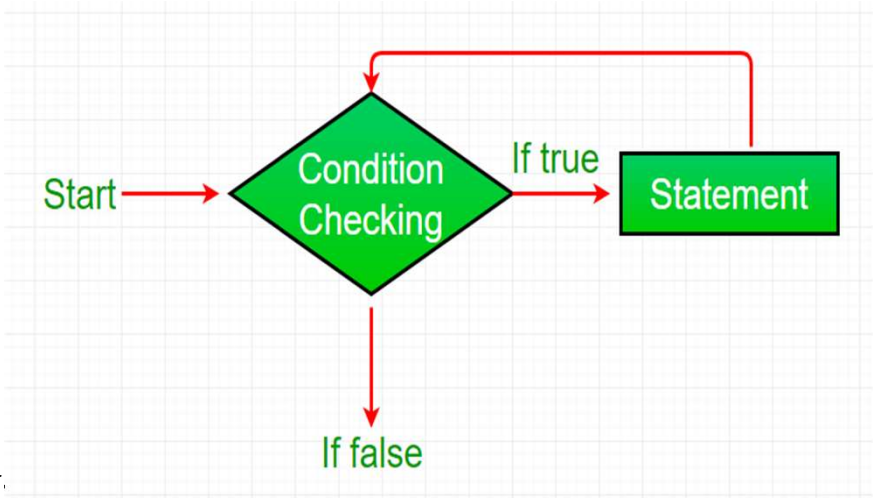
•while loop

•for loop

•do while loop

while loop

while loop can be addressed as an **entry control** loop. A while loop is a control flow statement that allows code to be executed repeatedly based on a given Boolean condition.



It is completed in 3 steps:

- Variable initialization.(e.g int x = 0;)
- condition(e.g while(x <= 10))
- Variable increment or decrement (x

```
variable initialization;
while(condition)
{
    statements;
    variable increment or decrement;
}
```

Example

Example: Program to print first 10 natural numbers

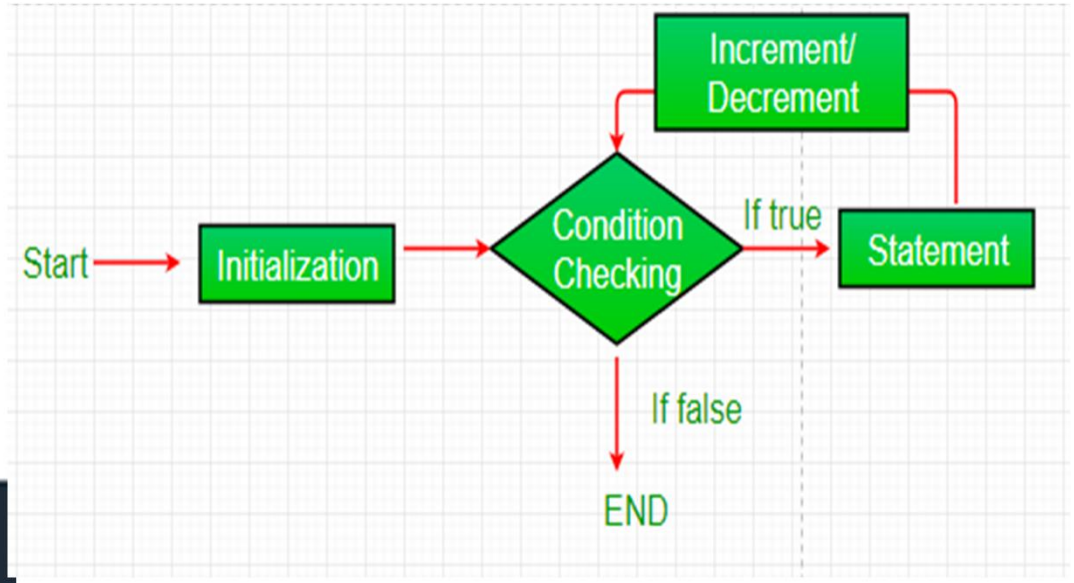
```
#include<stdio.h>

void main( )
{
    int x;
    x = 1;
    while(x <= 10)
    {
        printf("%d\t", x);
        /* below statement means, do x = x+1, increment x by 1*/
        x++;
    }
}
```

OUTPUT:

```
1 2 3 4 5 6 7 8 9 10
```

for loop

•for loop is used to execute a set of statements repeatedly until a particular condition is satisfied. We can say it is an **open ended loop.**




```
for(initialization; condition; increment/decrement)
{
    statement-block;
}
```

Conti...

The for loop is executed as follows:

- It first evaluates the initialization code.
- Then it checks the condition expression.
- If it is **true**, it executes the for-loop body.
- Then it evaluate the increment/decrement condition and again follows from step 2.
- When the condition expression becomes **false**, it exits the loop.

Conti...

```c
#include<stdio.h>

void main( )
{
    int x;
    for(x = 1; x <= 10; x++)
    {
        printf("%d\t", x);
    }
}
```

OUTPUT:

```
1 2 3 4 5 6 7 8 9 10
```

Nested for loop

We can also have nested for loops, i.e one for loop inside another for loop. Basic syntax is,

```
for(initialization; condition; increment/decrement)
{
    for(initialization; condition; increment/decrement)
    {
        statement ;
    }
}
```

Conti...

```c
#include<stdio.h>

void main( )
{
    int i, j;
    /* first for loop */
    for(i = 1; i < 5; i++)
    {
        printf("\n");
        /* second for loop inside the first */
        for(j = i; j > 0; j--)
        {
            printf("%d", j);
        }
    }
}
```

OUTPUT:

```
1

21

321

4321

54321
```

for loop vs while loop

FOR LOOP	WHILE LOOP
Initialization may be either in loop statement or outside the loop.	Initialization is always outside the loop.
Once the statement(s) is executed then after increment is done.	Increment can be done before or after the execution of the statement(s).
It is normally used when the number of iterations is known.	It is normally used when the number of iterations is unknown.
Condition is a relational expression.	Condition may be expression or non-zero value.
It is used when initialization and increment is simple.	It is used for complex initialization.
For is entry controlled loop.	While is also entry controlled loop.
for (init ; condition ; iteration) { statement(s); }	while (condition) { statement(s); }

do while loop

•In some situations it is necessary to execute body of the loop before testing the condition.

•Such situations can be handled with the help of do-while loop.

•do statement evaluates the body of the loop first and at the end the condition is checked using while statement.

•It means that the body of the loop will be executed at least once, even though the starting condition inside while is initialized to be **false**.

Conti...

```
do
{
    .....
    .....
}
while(condition)
```

Example

```c
#include<stdio.h>

void main()
{
    int a, i;
    a = 5;
    i = 1;
    do
    {
        printf("%d\t", a*i);
        i++;
    }
    while(i <= 10);
}
```

OUTPUT:

```
5 10 15 20 25 30 35 40 45 50
```

Jumping Out of Loops

Sometimes, while executing a loop, it becomes necessary to skip a part of the loop or to leave the loop as soon as certain condition becomes **true**.

1) break statement

When break statement is encountered inside a loop, the loop is immediately exited and the program continues with the statement immediately following the loop.

Syntax

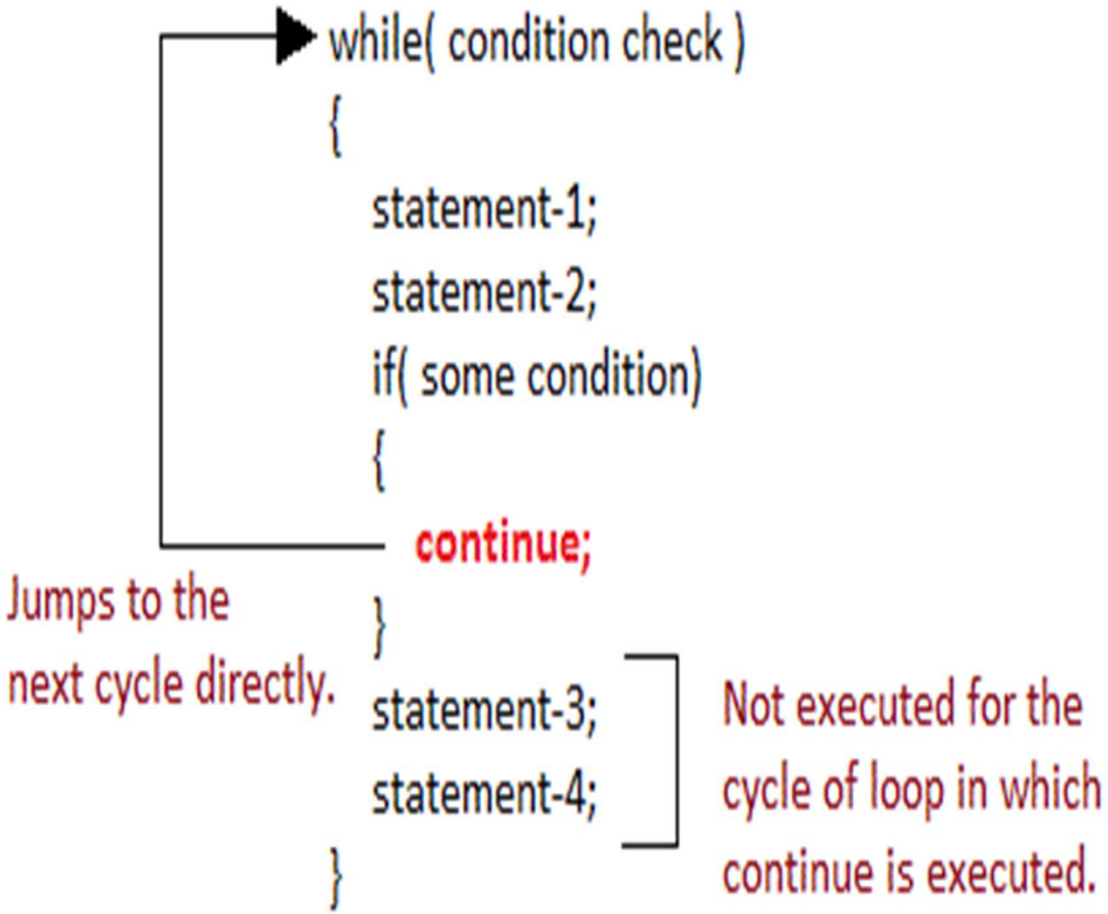
```
while( condition check )
{
   statement-1;
   statement-2;
   if( some condition)
   {
      break;
   }
   statement-3;
   statement-4;
}
```

Jumps out of the loop, no matter how many cycles are left, loop is exited.

2) continue statement

•It causes the control to go directly to the test-condition and then continue the loop process.

•On encountering continue, cursor leave the current cycle of loop, and starts with the next cycle.


while(condition check)
{
statement-1;
statement-2;
if(some condition)
{
continue;
}
statement-3;
statement-4;
}
Jumps to the next cycle directly.
Not executed for the cycle of loop in which continue is executed.

Assessment 1

1. Write about Decision making Statements?

Ans : _____

2. Write about Looping statements?

Ans : _____

References

TEXT BOOKS

1. Brian W. Kernighan and Dennis M. Ritchie, “The C Programming Language”, 2nd Edition, Pearson Education, 1988.

REFERENCES

1. Balagursamy “Programming In Ansi C “TATA Mc Graw Hill, First Edition,”2019.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L.Rivest, Clifford Stein, “Introduction to Algorithms", Second Edition, Mcgraw Hill, 2002.
3. Ashok.N.Kamthane,“ Computer Programming”, Pearson Education (India) (2010). (UNIT –II, III IV, V)

Thank You