

*

LOOPS

⇒

Set of instructions into the compiler to execute set of statements until the condition become false it is called loop.

⇒

Basic Purpose of loop is code repetition

⇒

In implementation when the repetitions are required then recommended to go for loops.

⇒

Generally iterative statements are called loop bcz way of the repetition forms a circle.

⇒

In 'C' prog. lang., loops are classified into 3 types

- 1) while loop
- 2) for loop
- 3) do-while loop

(1)

While loop

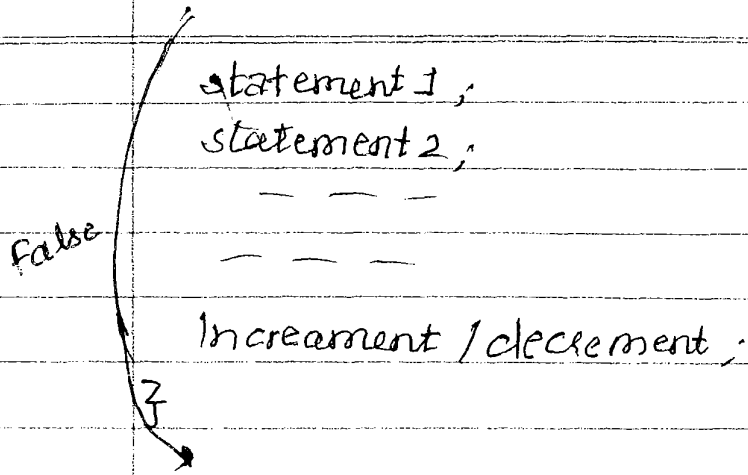
- when we are working with while loop pre-checking process is occurred i.e before execution of statements, block condition part is executed.
- while loop always repeats in clock direction.

Syntax :-

```

Assignment;
while (condition)
{
}
True

```



→ Acc. to syntax "while" cond. is true then control will pass within the body.

→ After execution of the body, once again control will pass back to the condⁿ and until the condition become false body will be repeated n no. of times.

→ When while condition become false, then control will pass outside of the body, if condⁿ is not false, then it becomes an infinite loop.

```
void main()
{
    int i;
    i = 1;
    while (i <= 10)
    {
        printf ("%d", i);
        i = i + 2;
    }
}
```

O/P: 1 3 5 7 9

* for loop :-

When we are working with for loop it contains 3 parts -

- 1) Initialisation
- 2) Condition
- 3) Iteration

Syntax :- for (initialization ; condition ; iteration)
 }
 statement block ;
 }

- When we are working with for loop, always execution process will start from initialisation.
- Initialisation part will be executed only once when we are passing the control within the body first time.
- After execution of initialisation part, control will pass to condition, if condn evaluated is true, then control will pass to statement block.
- After execution of statement block, control will pass to iteration, from iteration once again it will pass back to condition.
- Always repetition will come b/w condn, statement block and iteration only.
- When we are working with for loop, everything is optional but mandatory to place 2 semicolons.

while () → error

for (; ;) → valid

- When the condition part is not given in for loop, then it repeats infinite times bcz condn part is replaced with non-zero value (True value).
- When we are working with for loop, it repeats in anticlock direction.
- Always prechecking process occurs when we are working with for loop i.e. by execution of statement block condn part is executed.

while (0) → No repetition

for (; 0, ;) → It's it will repeat.

- In for loop, when a condn part is replaced with constant 0 then it repeats once, bcz no. of instances became 1 at the time of compilation.

int i;

i = 0;

while (i) → No repetition

for (; i;) → No rep.

↳ Condⁿ part shud be const 0
or NULL not a variable.

Prog:

```
void main()
{
```

```

int i;
for (i = 1; i <= 10; i = i + 2)
printf ("%d", i);
}

```

O/P: 1 3 5 7 9

3) do-while :-

- In implementation when we required to repeat the statement block atleast once then go for do-while loop.
- When we are working with do-while loop, post checking process occurs, i.e. after execution of statement block, condⁿ part is executed.
- When we are working with do-while loop, it repeats in clock direction.

Syntax :

```

Assignment ;
do
{
statement 1;
statement 2;
statement 3;
...
inc/dec ;
} while (condition);

```

- Acc. to syntax, semicolon must be required at the end of the body.