1) The while loop

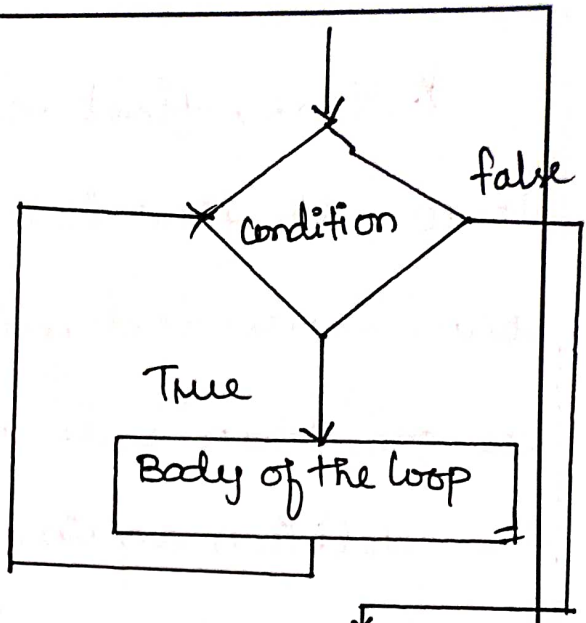
* It is a repetitive control structure, is used executes the statements with in the body until the condition becomes false.

* The while loop is an entry controlled loop statement, means the condition is evaluated first and it is true, then the body of the loop is executed.

* After executed of the body, the condition is once again evaluated and if it is true, the body is executed once again, the process of repeated execution of the body of the loop continues until the condition finally becomes false and the control is transferred out of the loop.

SYNTAX:

```
while (condition)
{
  .....
  body of the loop;
  .....
}
```



The body of the loop may have one or more statements, the blocking with the braces are needed only if the body contains two or more statements.

EXAMPLE:- Addition of numbers upto 10

```
main()
{ int i=1, sum=0;
  while (i<=10)
    {
      sum = sum+i;
      i++;
    }
  printf("The Sum of nos upto 10 is %d",
                                  sum);
}
```

// The while loop is a entry-Controlled loop or Top-Tested loop //

OUTPUT:-

The sum of numbers upto 10 is 55.

* Here, first check whether 'i' is less than or equal to 10. At first i=1 so the condition is true. now sum is added with the value of 'i' and get sum as one, than 'i' is incremented by 1, then it checks the condition again.

* This process is going on till the condition is false. when the condition is false the printf() stmt is executed and display the result.

2) THE DO-WHILE LOOP:

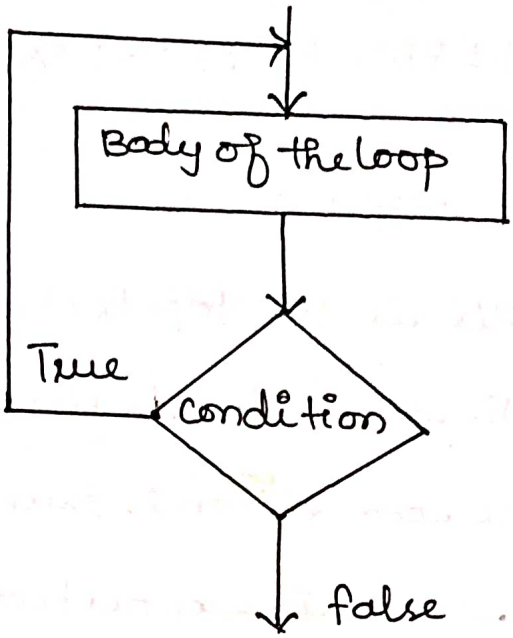
* The while loop makes a test of condition before the loop is executed. Therefore, the body of the loop may not be executed at all, if the condition is not satisfied at first attempt. In some situations it may be necessary to execute the body of the loop before the test condition is performed, such a situation the do....while loop is useful.

* It is also repetitive control structure and executes the body of the loop once irrespective of the condition, then it chicks the condition and

continues the execution until the condition becomes false.

SYNTAX:-

```
do
{
  .....
  body of the loop;
  .....
} while(condition);
```



Here, the statements with in the body of the loop is executed once, then it checks for the condition, if it is true, then it executes body until the condition becomes false

EXAMPLE:- Addition of numbers upto 10.

```
main()
{
  int i=1, sum=0;
  do
  {
    sum = sum+i;
    i++;
  }
  while(i<=10);
  printf("sum of the nos upto 10 is %d",sum);
}
```

OUTPUT:-

Sum of the numbers up to 10 is... 55

DIFFERENCE BETWEEN WHILE & DO...WHILE STATEMENTS:

WHILE	DO...WHILE
1) This is the top tested loop.	This is the bottom tested loop
2) The condition is first tested, if the condition is true then the block is executed until the condition becomes false.	It executes the body once, after it checks the condition if it is true the body is executed until the conditio false.
3) loop is not executed if the condition is false.	loop is executed atleast once even through the condition is false.
4) It has no termination.	It has the termination.

The do....while loop is a Exit-control loop.
or Bottom tested loop.

4) FOR LOOP:

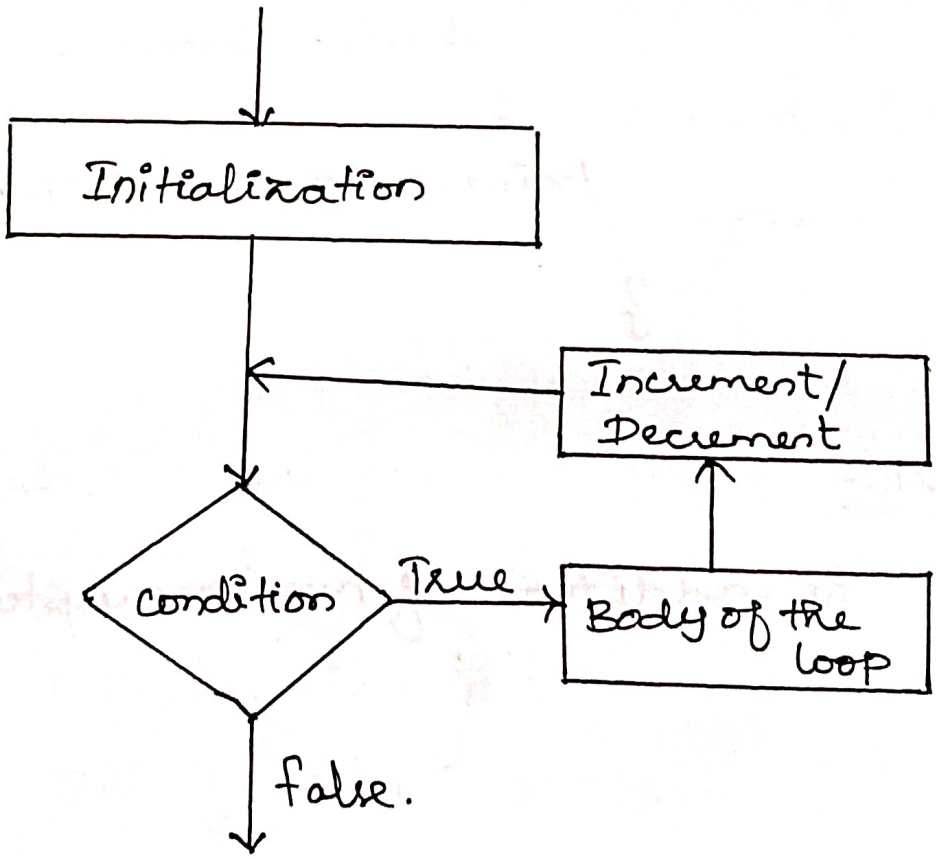
* The for loop is another repetitive control structure, and is used to execute set of instruction

repeatedly until the condition becomes false.

* The assignment, incrementation or decrementation and condition checking is done in for statement only, where as other control structures are not offered all these features in on statement.

SYNTAX:-

```
for (initialise counter; test condition;
                          increment/decrement
                          counter)
{
 ....
 body of the loop;
 .....
}
```



for loop has three parts

i) Initialise counter is used to initialize counter variable.

ii) Test condition is used to test the condition.

iii) Increment / decrement counter is used to increment or decrement counter variable.

EXAMPLE:- Addition of number upto 10.

```
main()
{
  int i, sum=0;
  for (i=1; i<=10 ; i++)
    {
      sum = sum+i;
    }
  Printf("the addition of nos upto
                          10 is %d", sum);
}
```

OUTPUT:-

The addition of numbers upto 10 is 55.

Note that the initialisation, testing and incrementation of loop is done in the for statement itself. Now how the above program is written in different way is given below

```
main ()
{
  int i;
  for (i=1; i<=10)
      {
       printf ("%.d\n", i);
       i= i+1;
      }
}
```

Here the incrementation is done with in the body of the for loop and not in the for statement. Note that inspite of this the Semicolon after the condition is necessary.

```
main ()
 {
   int i;
   for (i=0 ; i++<10;)
    printf ("%.d\n", i);
 }
```

* Here the comparison as well as the incrementation is done through the same statement i++<10. Since the '++' operator adds after 'i' firstly comparison is done, followed by incrementation. Note that it is necessary to initialise 'i' to 0.

NESTING OF FOR LOOPS:

* Like if statement for loop also needed. The loop with in the loop is called nested loop.

* To understand how nested loops work, look at the program given below

```
main()
{
  int i, j;
  for(i=1; i<=3; i++)
      {
       printf("\n");
       for(j=1; j<=3; j++)
       printf("%d", j);
      }
}
```

OUTPUT:

```
1  2  3
1  2  3
1  2  3
```

* Here for each value of 'i' the inner loop is executed completely, with in the inner loop 'j' taking from 1 to 4.

* The inner loop terminates when the value of 'j' is greater than 4. where 'in' causes the control to the next line for each entry of the 'i' loop.