

- After execution of the body, the test condition is once again evaluated and if it is true, the body is executed again.
- This process of repeated execution of the body continues until the test-condition finally becomes false and the control is transferred out of the loop.
- On exit, the program continues with the statement immediately after the body of the loop.
- The body of the loop may have one or more statements. The braces are needed only if the body contains two or more statements.

Pgm:

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

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

```
void main()
```

```
{ int sum=0, n=1;
```

```
while (n<=5)
```

```
{ sum = sum + n * n;
  n = n + 1; }
```

```
printf("Sum = %d\n", sum);
```

```
getch(); }
```

O/p
55

Step 1: sum = 14, n = 4
while (4 <= 5) true
sum = 14 + 4 * 4
sum = 14 + 16 = 30
n = 4 + 1 = 5

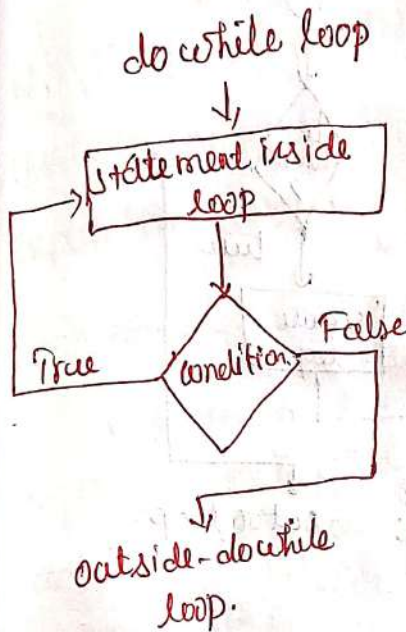
Step 2: sum = 30, n = 5
while (5 <= 5) true
sum = 30 + 5 * 5
sum = 30 + 25 = 55
n = 5 + 1 = 6
n = 6
sum = 55
while (6 <= 5) false

Do Statement: → while → advanced condition. before display.

→ To execute the body of the loop the test is performed.
do
body of the loop; → Such situations can be handled with the help of the do statement.

while (test-condition);

Working of do-while loop:



①- Program code (inside the body of loop) is executed once, before checking the condition. This is an exit-controlled loop.

② If the test expression is true, body of loop will be executed again & again, until the expression become false.

Pgm:

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

```
int main()
```

```
{ int a=10;
```

```
do {
```

```
printf("Value of a: %d \n", a);
```

```
a=a+1;
```

```
} while (a < 20);
```

```
return 0;
```

```
}
```

o/p:

Value of a: 10

...

③ For Statement:

→ The for loop is another Entry controlled loop.
that provide more ^{short} condise loop ctrl. structure
(initialization; test-condition; increment).

{ body of the loop
}

Sample: p.m:

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

```
int main()
```

```
{ int x;
```

```
for (x=0; x<=9; x=x+1)
```

```
{ printf("Value of x : %d\n", x);
```

```
}
```

```
return 0;
```

```
}
```

O/p:

Value of x : 0

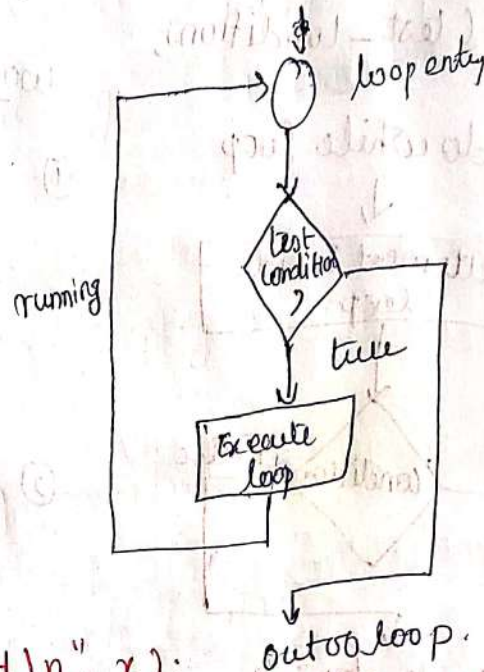
Value of x : 1

Value of x : 2

⋮

⋮

Value of x : 9



Execution of the for statement:

1. Initialization of the control variables is done by using assignment statement such as $i = 1$ & $count = 0$. The variables i and $count$ are known as loop-control variables.

2. The value of the control variable is tested using the test-condition. The test-condition is a relational expression such as $i \leq 10$ that determines when the loop will exit.

If the ~~loop~~ condition is true, the body of the loop will be executed. otherwise the loop is terminated and the execution continues with the statement the immediately follows the loop.