# Lecture - 16

## Programming in C

## Flow of Control (Part 04)

for →

## (B) While Loop :

↳ Conditional loop ( Condition is first priority)

* Explicitly mention the iteration and initialization

Syntax

```
initialise the iterative variable (i)
while (condition) {
        Block of while ;
        iteration ;
}
```

initialise the iterative variable (i) ↳ optional

iteration ; ← optional

loop

↳ कहाँ से चलेगा (Start Point)

→ Condition ( कब रोकना है )

→ iteration → Update

i → Condition

WAP to print first 10 positive integers. using 'while' loop.

```
int i = 0;
while( i < 11 ){
    printf(" %d", i);
    i++;
}
```

Start Point

Because the Value of i will remain '0' if we do not Update the Value of 'i'

$i = 0 \not\rightarrow 1 \not\rightarrow 2 \not\rightarrow 3$

| | | o/p |
|---|---|---|
| ① | $i<11$ (True) $\Rightarrow$ <br> $i++ \rightarrow i = 1$ | 0 |
| ② | $i<11$ (True) $\Rightarrow$ <br> $i++ \rightarrow i = 2$ | 1 |
| ③ | $i<11$ (True) $\Rightarrow$ <br> $i++ \Rightarrow i = 3$ | 2 |
| ④ | $i<11$ (True) $\Rightarrow$ | 3 |
| ⋮ | ⋮ | ⋮ |
| ⑩ | $i=9$ <br> $i<11$ (True) $\Rightarrow$ <br> $i++ \rightarrow i = 10$ | 9 |

$10 < 11$

⑪ $i<11$ (True) $\Rightarrow$ 10

$i++ \rightarrow i = 11$

⑫ $i<11 = 11<11$ (False)

⇓

Terminate

# Properties of while loop:

a) It is a Conditional loop, totally based on Condition

b) iteration is optional.

c) Check the Condition first then apply.

d) Initialization variables are explicitly mentioned
   - Initialization: mentioned outside of the loop
   - explicitly: Not the part of while loop

e) Risk of Infinite loop

Ex
```
while(1){
    printf("Hello");
}
```
1 → true; infinite

Write a program to print a multiplication table of any number using while loop.

input ← number (int), int i = 1

```
while (i<11){
    print → num x i ;
}
```

num → 12 x 1 = 12

12 x 2 = 24

⋮

12 x 10 = 120

12 → Common, 10 → Δi, 120 → result (num x i)

whileloop.c

```c
#include<stdio.h>

int main(){
    int num, i;
    printf("Enter a number: \n");
    scanf("%d", &num);
    i = 1;
    while (i<11){
        printf("%d x %d = %d\n",num,i, num*i);
        i++; // increment the value of i
    }
    return 0;
}
```

```
Enter a number:
13
13 x 1 = 13
13 x 2 = 26
13 x 3 = 39
13 x 4 = 52
13 x 5 = 65
13 x 6 = 78
13 x 7 = 91
13 x 8 = 104
13 x 9 = 117
13 x 10 = 130
PS C:\Users\sagar
 Chand\Programmin
```

## ③ do - while loop :

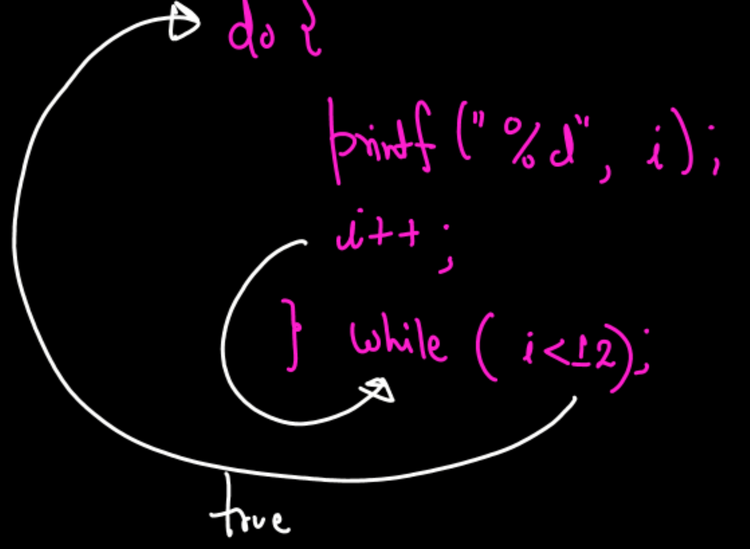
Same as while loop, but it execute the block at least one time.

⇒ on first attempt, the block executes before checking the condition.

```
do {
    // Block of do while loop
} while(Condition);
```

Ex

```
int i = 10;
do {
    printf("%d", i);
    i++;
} while (i<12);
```




i = 10

| | o/p |
|---|---|
| ① | 10 |

i++, i = 11

i < 12 ? (True)

| | |
|---|---|
| ② | 11 |

i++ → i = 12

i < 12

(12 < 12) ⇒ false

↳ Terminate

What will be the output of following Program?

(A)

```
int i = 20;
do{
    printf("Hello %d", i);
    i+1;
} while( i<20);
```

← false

(1)

i = 20

i++ ⇒ i = 21

O/P

Hello 20

i<20, 21<20 (False)

⇓

Exit the loop

(B)

```
int i = 3;
do {
    printf( "%d", i);
    i--;
} while( i<1);
```

← false

i = 3

(1) i = 3

i-- ⇒ i = 2

O/P

3

(2) i<1 (2<1) False

↓

Exit the loop

whileloop.c ● dowhileloop.c ×

dowhileloop.c > main()

```c
#include<stdio.h>
int main(){
    int i = 10;
    do {
        printf("Hello%d\n", i);
        i++;
    } while(i<5); //false hai starting se
    return 0;
}
```

```
Hello10
PS C:\Users
 Chand\Prog
```

Ex

⇒ factorial

⇒ $^{n}C_{r}$ / $^{n}P_{r}$

⇒ Kaprekar's Constant

→ prime No.

→ Binary / NOT

→ Decimal → other BS

→ other BS → Decimal

→ Bank

→ fibonacci

→ Ramanuj

tomorrow → 2 Oct → DPP → 2-3 hrs

Conditional + loop = Program

→ MCQ series ⇒ 100 MCQ

1 Oct →

⇒ Nested loop → loop inside the loop ← matrix (Advance) ← funct$^{n}$

↳ Pattern