Loop

The looping can be defined as repeating the same process multiple times until a specific condition satisfies.

Types of C Loops

- do while
- while
- for

Do while

Do-while loop:

The do-while loop continues until a given condition satisfies. It is also called post tested loop. It is used when it is necessary to execute the loop at least once.

The syntax of do-while loop:

```
Do
{
//code to be executed
}while(condition);
```

Program to print table for the given number using do while loop

```
#include<stdio.h>
int main(){
int i=1,number=0;
printf("Enter a number: ");
scanf("%d",&number);
do{
printf("%d \n",(number*i));
i++;
}while(i<=10);
return 0;
}
```

Infinitive do while loop

The do-while loop will run infinite times if we pass any non-zero value as the conditional expression.

```
Do
{
//statement
}while(1);
```

While Loop

While Loop:

While loop is also known as a pre-tested loop. In general, a while loop allows a part of the code to be executed multiple times depending upon a given boolean condition. It can be viewed as a repeating if statement. The while loop is mostly used in the case where the number of iterations is not known in advance.

Syntax of while loop:

```
while(condition)
{
//code to be executed
}
```

Example of While Loop

```
#include<stdio.h>
int main(){
int i=1;
while(i<=10){
printf("%d \n",i);
i++;
}
return 0;
}
```

Infinitive while loop

If the expression passed in while loop results in any non-zero value then the loop will run the infinite number of times

```
while(1)
{
//statement
}
```

For Loop

The **for loop in C language** is used to iterate the statements or a part of the program several times. It is frequently used to traverse the data structures like the array and linked list.

Syntax of for loop:

```
for(Expression 1; Expression 2; Expression 3)
{
//code to be executed
}
```

Example of For Loop

```
#include<stdio.h>
int main(){
int i=0;
for(i=1;i<=10;i++){
printf("%d \n",i);
}
return 0;
}
```

Infinitive for loop

```
#include<stdio.h>
void main ()
{
    for(;;)
    {
        printf("welcome to infinite loop");
    }
}
```

Nested for loop

The nested for loop means any type of loop which is defined inside the 'for' loop.

```
for (initialization; condition; update)
{
    for(initialization; condition; update)
    {
        // inner loop statements.
    }
    // outer loop statements.
}
```

Example Nested for loop

```
#include <stdio.h>
int main()
{
   int n;// variable declaration
   printf("Enter the value of n :");
   // Displaying the n tables.
   for(int i=1;i<=n;i++)  // outer loop
   {
       for(int j=1;j<=10;j++)  // inner loop
       {
         printf("%d\t",(i*j)); // printing the value.
       }
        printf("\n");
 }
```

Lab Task:
Write a C program to count number of digits in any number.

```
#include<stdio.h>
int main()
{
long long num;
int count =0;
printf(“Enter any number”);
scanf (“%lld”, &num);
Do
{
/* Increment digit cout */
count++
num/=10;
} while(num!=0);
printf(“Total digitd: %d”, count);
return 0;
```

Write a C program to enter any number and print its reverse.

```
#include<stdio.h>
int main()
//input  n=234 then output will be 432
int n,r;
Printf(“Enter Number: “);
scanf(“%d”,&n);
While(n>0)
{
r=n%10;
printf(“%d”,r);
n=n/10;
```

Write a C Program to calculate sum of a series

Sum of the series 1 + 2 + 3 + … + n = n (n+1)/2

```
#include<stdio.h>
int main() {
int n,i;
int sum=0;
printf("Enter the max values of series n  ");
scanf("%d", &n);
sum = (n * (n + 1)) / 2;
printf("Sum of the series: ");
for (i =1;i <= n;i++) {
if (i!=n)
printf("%d + ",i); else
printf("%d = %d ",i,sum);
}
return 0;
```

Write a C program to enter any number and calculate its factorial.

```
#iinclude <stdio.h>
 int main()
{
  int i,f=1,num;
  printf("Input the number : ");
  scanf("%d",&num);
  for(i=1;i<=num;i++)
  {
        f=f*i;
   }
 printf("The Factorial of %d is: %d\n",num,f);
}
```