रा.इ.सू.प्रौ.सं NIELIT

NIELIT GORAKHPUR

Course Name: O Level (2nd Sem B2 and B3 Batch) **Subject: C Language**
Topic: Break and Continue Statement **Date: 31-March-2020**

Break Statement :
Sometimes, it is necessary to exit immediately from a loop as soon as the condition is satisfied. When break statement is used inside a loop, then it can cause to terminate from a loop. The statements after break statement are skipped.
This can also be used in switch case control structure. Whenever it is encountered in switch-case block, the control comes out of the switch-case.

Syntax: break;

```
while(condition)
{
- - - - -
-    - - - - -
break;
- - -
- - -
}
```

Example of Break Statement:

```
#include <stdio.h>
int main()
{
     int x;
     for (x=10; x>=1; x--)
     {
         printf("x: %d\n", x);
         if (x = = 9)
         {
            break;
         }
     }
    printf("Out of loop");
    return 0;
}
```

Output:

```
x: 10
x: 9
Out of loop
```

Continue Statement :
Sometimes, it is required to skip a part of a body of loop under specific conditions. So, C supports 'continue' statement to overcome this problem. The working structure of 'continue' is similar as that of that break statement but difference is that it cannot terminate the loop. Continue statement simply skips statements and continues next iteration.

Syntax: continue;

```
while(condition)
{
- - - - -
-    - - - - -
Continue;
-    - - -
-    - - -
}
```

Example of Continue Statement:

```
#include <stdio.h>
void main ()
{
  int a = 11;
  do {
   if( a == 15)
   {
       a = a + 1;
       continue;
   }
  printf("value of a: %d\n", a);
  a++;
} while( a < 20 );
}
```

Output:

```
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 16
value of a: 17
value of a: 18
value of a: 19
```

Example 1: Program to input any number and check the number is prime number or not prime number.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int x,y=2;
clrscr();
printf("Enter any number\n");
scanf("%d",&x);
while(x>y)
{
if(x%y==0)
{
printf("Not prime");
break;
}
else
y++;
}
if(y==x)
printf("Prime number");
getch();
}
```

Example 2: Program to calculate the sum of a 10 numbers. If a negative number is entered, the loop terminates.

```
# include <stdio.h>
void main()
{
   int i,number, sum = 0;
   for(i=1; i <= 10; i++)
   {
      printf("Enter any number ");
      scanf("%d",&number);
      if(number < 0)
      {
         break;
      }
      sum += number;
   }
   printf("Sum = %.d",sum);
}
```

Example 3: Program to calculate the sum of a 10 numbers. If a negative number is entered, the loop skip the statement.

```
# include <stdio.h>
void main()
{
   int i, number, sum = 0;
   for(i=1; i <= 10; i++)
   {
      printf("Enter any number: ");
      scanf("%d",&number);
      if(number < 0)
      {
            continue;
        }
        sum += number;
    }
    printf("Sum = %.2lf",sum);
}
```

Try Yourself:

Q.1. Rewrite the Example 2 with the help of while loop.

Q.2. Rewrite the Example 3 with the help of while loop.

Q.3. Write a program to print only two Armstrong number b/w 1 to 500.

Q.4. Write a program to print all the prime number b/w 1 to 100.