CONTENT 11

Break and Continue Statements In C

#### Break Statement :

* Break statement is used to break the loop or switch case statements execution and brings the control to the next block of code after loop or switch case.
* Break statements are used to bring the program control out of the loop.
* The break statement is used inside loops or switch statement in C Language.

**Code 1:**

// Program to calculate the sum of numbers (10 numbers max)

// If the user enters a negative number, the loop terminates

#include<stdio.h>

int main(int argc, char const \*argv[])

{   int i;

    double number,sum=0.0;

    for ( i =1; i <=10; i++)

    {

        printf("Enter n%d \n",i);

        scanf("%lf", &number);

         //if user input neg no then break the loop

    if (number<0.0)

    {

        break;

    }

    sum += number;  //sum= sum + number

    }

    printf("the sum is %.4lf",sum);

    return 0;

}

**Output:**

1. **Enter n1**

**2**

1. **Enter n2**

**4**

1. **Enter n3**

**2**

1. **Enter n4**

**-4**

1. **the sum is 8.0000**

As you see the code gives sum 8.0000 this means code stops at line 4 and gives the result because we said that if value is less than or equal to 0.0 then break the execution(don’t stop).

printf("the sum is %.4lf",sum);

here %.4 print the 4 decimal after ‘.’point. (as you see in result)

### Continue Statement :

* The **continue statement** is used inside [loops](https://beginnersbook.com/2014/01/c-loops-examples/) in C Language. When a continue statement is encountered inside the loop, control jumps to the beginning of the loop for next iteration, skipping the execution of statements inside the body of loop after continue statement.
* It is used to bring the control to the next iteration of the loop.
* The continue statement skips some code inside the loop and continues with the next iteration.
* It is mainly used for a condition so that we can skip some lines of code for a particular condition.
* It forces next iteration in loop i.e. as break terminates the loop but continue forces the next iteration of the loop.

**Code 2 :**

// Program to calculate the sum of numbers (10 numbers max)

// If the user enters a negative number, the loop terminates

#include<stdio.h>

int main(int argc, char const \*argv[])

{   int i;

    double number,sum=0.0;

    for ( i =1; i <=10; i++)

    {

        printf("Enter n%d \n",i);

        scanf("%lf", &number);

         //if user input neg no then break the loop

    if (number<0.0)

    {

        break;

    }

    sum += number;  //sum= sum + number

    }

    printf("the sum is %.4lf",sum);

    return 0;

}

**Output:**

1. **Enter n1: 1**
2. **Enter n2: 4**
3. **Enter n3: 6**
4. **Enter n4: 7**
5. **Enter n5: -7**
6. **Enter n6: -7**
7. **Enter n7: 7**
8. **Enter n8: 1**
9. **Enter n9: 2**
10. **Enter n10: 1**
11. **The sum is 29.000**

As you see the code gives sum 29.000 this means code just skipped or ignore the line 5 and 6 because we said that if value is less than or equal to 0.0 then continue the execution(don’t stop).

**As you both the codes (code1 and code 2) you would also understand the difference between break and continue statements.**