



LAB REPORT

Submitted By,

Samin Sadik Khan

Department of CSE

Batch 60thD

ID: 0282220005101018

Submitted To,

Md. Alamgir Hossain

Senior Lecturer

Department of CSE

Course: CSE-132

Problem no. 01

Name of Problem: Write a C++ program applying the concept of OOP to count the factors of a number.

Submission Date: 01 March 2023

Problem No: 01**Problem Name:** C++ program to count the factors of a number (applying the concept of OOP)**Case Analysis:**

Case number	Inputs	Expected Outputs
01	0	0
02	18	06
03	307	02
04	96	12
05	1234	04

Code:

```
#include <iostream>
using namespace std;

class factor
{
public:
    int n;
    factor()
    {
        cout << "Enter a positive number: ";
        cin >> n;
    }
    void solve()
    {
        int Count = 0;
        if(n>0)
        {
            cout << "Factors of " << n << " are: ";
            for(int i=1; i*i<n; i++)
            {
                if(n%i==0)
                {
                    if(i*i==n) {
                        cout << i << " ";
                        Count++;
                    } else {
                        cout << i << " " << n/i << " ";
                        Count += 2;
                    }
                }
            }
            cout << endl;
        }
        cout << "The number of factors: " << Count << endl;
    }
};
```

```
int main()
{
    factor o;
    o.solve();
    return 0;
}
```

Input:

0, 18, 307, 96, 1234.

Result & Opinion:

The number of factors of 0 is 0.

The number of factors of 18 is 6, which are 1, 2, 3, 6, 9, 18.

The number of factors of 307 is 2, which are 1, 307.

The number of factors of 96 is 12, which are 1, 2, 3, 4, 6, 8, 12, 16, 24, 32, 48, 96.

The number of factors of 1234 is 4, which are 1, 2, 617, 1234.

```
Enter a positive number: 18
Factors of 18 are: 1 18 2 9 3 6
The number of factors: 6
```

According to the results we can observe that, the program is providing with the expected outputs. In this program, for every integer except 0, there are two lines of output. The first line displays all the factors of the number & the second line shows the total number of factors the input has. However, the display of the factors in general doesn't follow ascending or descending order, which was programmed this way to optimize the time complexity by only checking from 1 to ($\sqrt{n}-1$) instead of 1 to n in the loop.

```
Enter a positive number: 0
The number of factors: 0
```

In our analysis, 0 is a special case. Because the minimum number of factors for a positive number is 2, those are the number itself & 1. But 0 is neither a positive nor a negative number, it is a non-negative number. So, the output should be zero as well, which the program could successfully calculate with the general conditions in the code, but to avoid getting blank output in the displaying of the factors of 0 or any negative numbers, "if(n>0)" condition was used.

For a prime number such as 307, the output should be 2, which is the lowest possible number of factors for a positive number. So, we can conclude that the program works correctly and as intended.