



Air University (Multan Campus)

Dept. Computer Sciences, BSCS-V Fall'2017

Assignment# 2

TDD- Test Driven Development

Subject: Software Engineering

Submitted to:

Sir Ahmad Mohsin

Submitted by:

Umamah Ayyaz Ansari (153174)

Date of Submission:

25th September, 2017

Question:

Requirements

1. Factorial calculates the multiplication of the all positive numbers less than or equal Target Number (n).
2. Target Number (n) is a non-negative integer.

Unit Tests to be created:

Zero factorial is One $0! = 1$

One factorial is One $1! = 1*0! = 1$

Two factorial is Two $2! = 2*1! = 2$

Three factorial is Six $3! = 3*2! = 6$

Steps:

Write Stubs as followings:

zerofactoriaisone()

onefactoriaisone()

twofactoriaistwo()

threefactoriaissix()

fivefactorialisonetwenty()

Code your Factorial Calculation Application at the end.

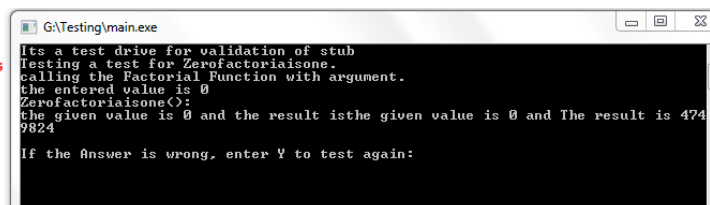
Solution:

Step# 1: Testing:

Test No.	Scenario	Input	Output	Pass/Fail	Remarks
1	The factorial of 0 is 1	0	4749824	Fail	Changes should be done in the stub "zerofactoriaisone(int)"

```
//performing a test for zerofactoriaisone()
do {
    cout << "Testing a test for zerofactoriaisone.\n";
    cout << "calling the Factorial Function with argument." << endl;
    result = zerofactoriaisone(0);
    cout << "the given value is " << 0 << " and ";
    cout << "The result is " << result << endl;
    cout << endl;

    cout << "If the Answer is wrong, enter Y to test again: ";
    cin >> choice;
}while(choice=='Y' || choice=='y');
```

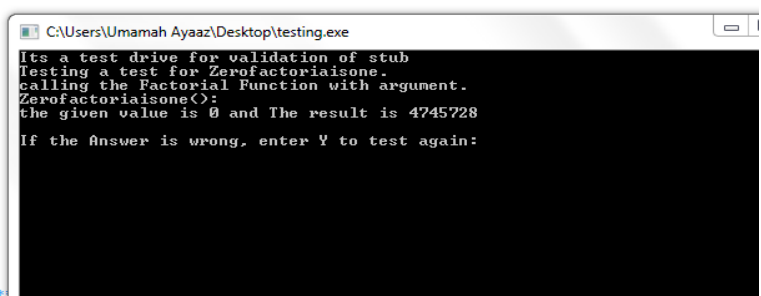


Test No.	Scenario	Input	Output	Pass/Fail	Remarks
2	The factorial of 0 is 1	0	4745728	Fail	Answer should be initialized in the stub "zerofactoriaisone(int)"

```
//*****
// FUNC: zerofactoriaisone()
// operation: to check if the factorial of zero is 1
// INPUT: 0 (initialized)
// OUTPUT: should be 1.

int zerofactoriaisone(int n)
{
    int result;
    for (int i=1; i<=n; i++)
    {
        result*=i;
    }
    cout << "zerofactoriaisone(): \n";

    if (result==1)
    {
        return result;
    }
}
//*****
```



Test No.	Scenario	Input	Output	Pass/Fail	Remarks
3	The factorial of 0 is 1	0	1	Pass	Initialization worked.

```
//performing a test for Zerofactoriaisone()
do {
    cout << "Testing a test for Zerofactoriaisone.\n";
    cout << "calling the Factorial Function with argument." << endl;
    cout << "the given value is "<< 0 << " and ";
    cout << "The result is " << Zerofactoriaisone(0) << endl;
    cout << endl;

    cout << "If the Answer is wrong, enter Y to test again: ";
    cin >> choice;
}while(choice=='Y' || choice=='y');
```

Test No.	Scenario	Input	Output	Pass/Fail	Remarks
4	The factorial of 1 is 1	1	4745728	Fail	Answer should be initialized in all stubs.

```
*****
// FUNC: onefactoriaisone()
// operation: to check if the factorial of 1 is 1
// INPUT: 1 (initialized)
// OUTPUT: should be 1.

int onefactoriaisone(int n)
{
    int result;
    for (int i=1; i<=n; i++)
    {
        result*=i;
    }
    cout << "onefactoriaisone(): \n";

    if (result==1)
    {
        return result;
    }
}
*****
```

Test No.	Scenario	Input	Output	Pass/Fail	Remarks
5	The factorial of 1 is 1	1	1	Pass	Initialization worked.

```
.....
//*****
// FUNC: onefactoriaisone()
// operation: to check if the factorial of 1 is 1
// INPUT: 1 (initialized)
// OUTPUT: should be 1.

int OneFactoriaisone(int n)
{
    cout << "In the Func: OneFactoriaisone. \n";
    cout << "The Factorial of " << n << " is ";

    if (n == 1)
        return 1;
}
//*****
```

Test No.	Scenario	Input	Output	Pass/Fail	Remarks
6	The factorial of 2 is 2	2	2	Pass	Answer is initialized

```

//*****
// FUNC: TwoFactoriaistwo()
// operation: to check if the factorial of 2 is 2
// INPUT: 2 (initialized)
// OUTPUT: should be 2.
int TwoFactoriaistwo(int n)
{
    cout << "In the Func: TwoFactoriaistwo. \n";
    cout << "The Factorial of " << n << " is ";

    if (n == 2)
        return 2;
}
//*****

```

```

calling the Factorial Function with argument.
the given value is 0 and In the Func: OneFactoriaisone.
The Factorial of 1 is The result is 1
If the Answer is wrong, enter Y to test again: n
Testing a test for TwoFactoriaistwo().
calling the Factorial Function with argument.
the given value is 0 and In the Func: TwoFactoriaistwo.
The Factorial of 2 is The result is 2
If the Answer is wrong, enter Y to test again:

```

Test No.	Scenario	Input	Output	Pass/Fail	Remarks
7	The factorial of 3 is 6	3	6	Pass	Answer is initialized

```

//*****
// FUNC: ThreeFactoriaissix()
// operation: to check if the factorial of 3 is 6
// INPUT: 3 (initialized)
// OUTPUT: should be 6.
int ThreeFactoriaissix(int n)
{
    cout << "In the Func: ThreeFactoriaissix. \n";
    cout << "The Factorial of " << n << " is ";

    if (n == 3)
        return 6;
}
//*****

```

```

the given value is 0 and In the Func: TwoFactoriaistwo.
The Factorial of 2 is The result is 2
If the Answer is wrong, enter Y to test again: n
Testing a test for ThreeFactoriaissix().
calling the Factorial Function with argument.
the given value is 0 and In the Func: ThreeFactoriaissix.
The Factorial of 3 is The result is 6
If the Answer is wrong, enter Y to test again:

```

Test No.	Scenario	Input	Output	Pass/Fail	Remarks
8	The factorial of 5 is 120	5	120	Pass	Answer is initialized

```

//*****
// FUNC: FiveFactoriaisonetwenty()
// operation: to check if the factorial of 5 is 120
// INPUT: 5 (initialized)
// OUTPUT: should be 120.
int FiveFactoriaisonetwenty(int n)
{
    cout << "In the Func: FiveFactoriaisonetwenty. \n";
    cout << "The Factorial of " << n << " is ";

    if (n == 5)
        return 120;
}
//*****

```

```

If the Answer is wrong, enter Y to test again: n
Testing a test for FiveFactoriaisonetwenty().
calling the Factorial Function with argument.
the given value is 0 and In the Func: FiveFactoriaisonetwenty.
The Factorial of 5 is The result is 120
If the Answer is wrong, enter Y to test again: n
-----
Process exited after 180.2 seconds with return value 0
Press any key to continue . . .

```

Test No.	Scenario	Input	Output	Pass/Fail	Remarks
9	The input should not be negative	-9	Accepted as positive number.	Fail	Limit should be applied for program to accept positive input only

```

int main ()
{
    int n;

    cout << "Enter a Positive number, \n";
    cin >> n;
    cout << endl;

    return 0;
}

```

Test No.	Scenario	Input	Output	Pass/Fail	Remarks
10	The input should not be negative	-9	Not Accepted as positive number.	Pass	Limitation worked.

```

cout << "Enter a Positive number, \n";
cin >> n;
cout << endl;
while (n<0)
{
    cout << "Invalid entry.\n";
    cout << "Enter a Positive number.\n";
    cin >> n;
}

return 0;

```

Test No.	Scenario	Input	Output	Pass/Fail	Remarks
11	The input should not be an Alphabet	k	Accepted as positive number.	Fail	Limit should be applied for program to accept positive number as input only

```

int n;

cout << "Enter a Positive number, \n";
cin >> n;
cout << endl;
while (n<0)
{
    cout << "Invalid entry.\n";
    cout << "Enter a Positive number.\n";
    cin >> n;
}

return 0;
}

```

Test No.	Scenario	Input	Output	Pass/Fail	Remarks
12	The input should not be an Alphabet	k	Not Accepted as positive number.	Pass	Limitation worked

```

cout << "Enter a Positive number: ";
cin >> n;
while (n<0 || cin.fail())
{
    cout << endl;
    cin.clear(); // clear input buffer to restore cin to a usable state
    cin.ignore(INT_MAX, '\n'); // ignore last input
    cout << "Invalid entry.\n";
    cout << "Enter a Positive number.\n";
    cin >> n;
}

return 0;

```

```

Enter a Positive number: a
Invalid entry.
Enter a Positive number.
-9
Invalid entry.
Enter a Positive number.
5
Process exited after 27.66 seconds with return value 0
Press any key to continue . . .

```

Test No.	Scenario	Input	Output	Pass/Fail	Remarks
13	The input should not be a Decimal number	9.5	Accepted as positive number.	Fail	Limit should be applied

```

cout << "Enter a Positive number: ";
cin >> n;
while (n<0 || cin.fail())
{
    cout << endl;
    cin.clear(); // clear input buffer to restore cin to a usable state
    cin.ignore(INT_MAX, '\n'); // ignore last input
    cout << "Invalid entry.\n";
    cout << "Enter a Positive number.\n";
    cin >> n;
}

return 0;

```

```

C:\Users\Umamah Ayaz\Desktop\New folder (2)\Tester2.exe
Enter a Positive number: 9.5
Process exited after 3.917 seconds with return value 0
Press any key to continue . . .

```

Test No.	Scenario	Input	Output	Pass/Fail	Remarks
14	The input should not be a Decimal number	9.5	Accepted as positive number, since data type was "int", the decimal part is ignored	Pass	

```

int n;

cout << "Enter a Positive number: ";
cin >> n;
while (n<0 || cin.fail())
{
    cout << endl;
    cin.clear(); // clear input buffer to restore cin to a usable state
    cin.ignore(INT_MAX, '\n'); // ignore last input
    cout << "Invalid entry.\n";
    cout << "Enter a Positive number.\n";
    cin >> n;
}

cout << "Input is " << n; // Decimal part is ignored due to data type "int"

return 0;

```

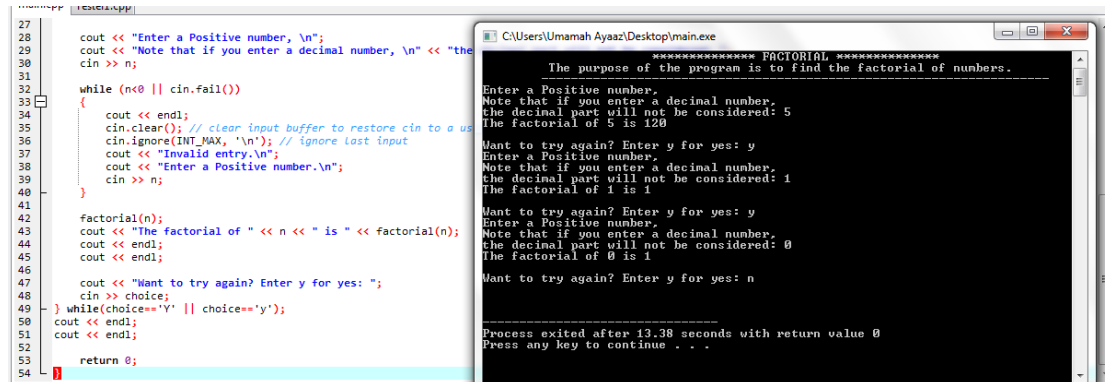
```

C:\Users\Umamah Ayaz\Desktop\New folder (2)\Tester2.exe
Enter a Positive number: 9.5
Input is 9
Process exited after 5.634 seconds with return value 0
Press any key to continue . . .

```

After the multiple test, code is finally ready to be written.

The “Factorial of the Positive Number”:



The image shows a C++ IDE with two windows. The left window displays the source code for a program that calculates the factorial of a positive number. The code includes input validation, a loop for repeated execution, and a function to calculate the factorial. The right window shows the program's execution output, which includes the purpose of the program, prompts for input, and the calculated factorial values for 5 and 1.

```
27 cout << "Enter a Positive number, \n";
28 cout << "Note that if you enter a decimal number, \n" << "the
29 cin >> n;
30
31
32 while (n<0 || cin.fail())
33 {
34     cout << endl;
35     cin.clear(); // clear input buffer to restore cin to a usable state
36     cin.ignore(INT_MAX, '\n'); // ignore last input
37     cout << "Invalid entry.\n";
38     cout << "Enter a Positive number.\n";
39     cin >> n;
40 }
41
42 factorial(n);
43 cout << "The factorial of " << n << " is " << factorial(n);
44 cout << endl;
45 cout << endl;
46
47 cout << "Want to try again? Enter y for yes: ";
48 cin >> choice;
49 } while(choice=='Y' || choice=='y');
50 cout << endl;
51 cout << endl;
52
53 return 0;
54
```

```
***** FACTORIAL *****
The purpose of the program is to find the factorial of numbers.
Enter a Positive number,
Note that if you enter a decimal number,
the decimal part will not be considered: 5
The factorial of 5 is 120
Want to try again? Enter y for yes: y
Enter a Positive number,
Note that if you enter a decimal number,
the decimal part will not be considered: 1
The factorial of 1 is 1
Want to try again? Enter y for yes: y
Enter a Positive number,
Note that if you enter a decimal number,
the decimal part will not be considered: 0
The factorial of 0 is 1
Want to try again? Enter y for yes: n

Process exited after 13.38 seconds with return value 0
Press any key to continue . . .
```