CIS 200 - Lab 0602

1. Problem Statement

Perform FACTORIAL !

2. Requirements

2.1 Assumptions

- User will only input integers
- User will only input into the command line

2.2 Specifications

- User will input integer to take factorial! of
- Program will print factorial!

3. Decomposition Diagram

- Program
    - Input
        - User inputs integer
    - Process
        - FACTORIAL!
    - Output
        - Answer

4. Test Strategy

- Valid Data
- Invalid Data

5. Test Plan Version 1

Test Strategy	#	Description	Input	Expected Output	Actual Output	Pass/Fail
Valid Data	1	Valid Factorial				
Valid Data	2	Valid Factorial				
Invalid Data	1	Invalid Factorial				

Invalid Data	2	Stack Overflow				

6. Initial Algorithm

1. Create Function
    a. Name: factorial
    b. Parameters: integer input, bool methodStatus
    c. Return: integer answer
    d. Method:
        i. Set methodStatus to true
        ii. If input is less than 0
            1. Set methodStatus to false
            2. Return -1
        iii. Else If input equals 0
            1. Return 1
        iv. Else
            1. Return Input * factorial(input - 1, methodStatus)
2. Create Main
    a. Delcare methodStatus as false
    b. Declare integer input as 0
    c. Prompt user for input
    d. Set input
    e. Perform Factorial with input

7. Test Plan Version 2

Test Strategy	#	Description	Input	Expected Output	Actual Output	Pass/Fail
Valid Data	1	Valid Factorial	5	120		
Valid Data	2	Valid Factorial	0	1		
Invalid Data	1	Invalid Factorial	-1	-1		
Invalid Data	2	Stack Overflow		Stack Overflow		

8. Code

```
//Program Name: FACTORIAL
//Programmer Name: Arthur Aigeltinger IV
//Description: Use recursion to perform the FACTORIAL !
```

```
//Date Created: 10/23/18

#include <iostream>

//Function Prototypes
int factorial(int input, bool &methodStatus);

int main()
{
	//Declare Variables
	bool methodStatus = false;
	int input = 0;

	std::cout << "Please enter an integer to find the factorial of: ";
	std::cin >> input;

	std::cout << input << "! = " << factorial(input, methodStatus) << std::endl;


	system("pause");
	return 0;
}

int factorial(int input, bool & methodStatus)
{
	methodStatus = true;

	if (input < 0)
	{
		methodStatus = false;
		return -1;
	}
	else if (input == 0)
	{
		return 1;
	}
	else
	{
		return (input * factorial(input - 1, methodStatus));
	}
}
```

9. Updated Algorithm

1. Create Function
    a. Name: factorial
    b. Parameters: integer input, bool methodStatus

c. Return: integer answer
d. Method:
    i. Set methodStatus to true
    ii. If input is less than 0
        1. Set methodStatus to false
        2. Return -1
    iii. Else If input equals 0
        1. Return 1
    iv. Else
        1. Return Input * factorial(input - 1, methodStatus)

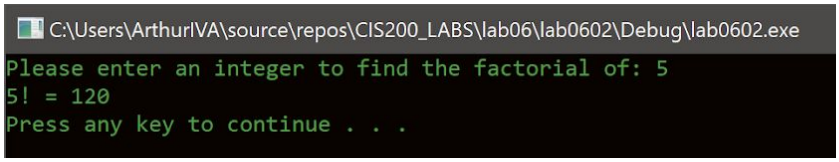
2. Create Main
    a. Delcare methodStatus as false
    b. Declare integer input as 0
    c. Prompt user for input
    d. Set input
    e. Perform Factorial with input

10. Test Plan Version 3

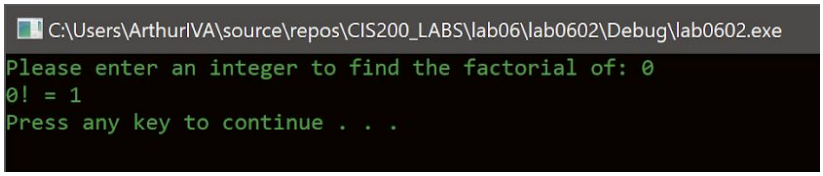
Test Strategy	#	Description	Input	Expected Output	Actual Output	Pass/Fail
Valid Data	1	Valid Factorial	5	120	120	Pass
Valid Data	2	Valid Factorial	0	1	1	Pass
Invalid Data	1	Invalid Factorial	-1	-1	-1	Pass
Invalid Data	2	Stack Overflow	50000	Stack Overflow	Stack Overflow	Pass

11. Screenshots

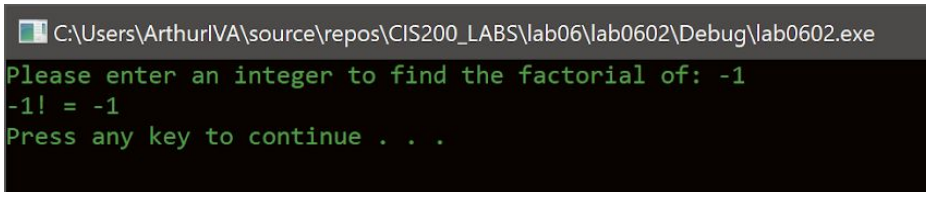
Valid Test Case 1



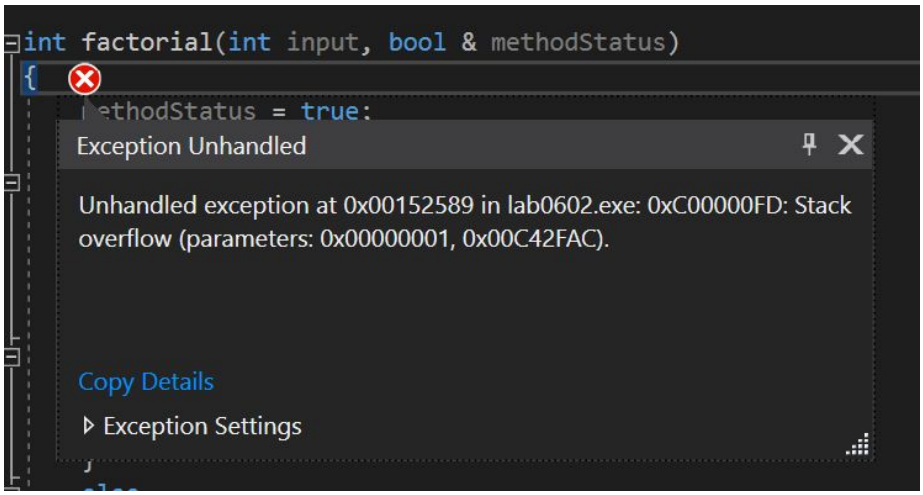
Valid Test Case 2



Invalid Test Case 1



Invalid Test Case 2



12. Error Log

Error Type (Logic/Runtime)	Cause of Error	Solution to Error
N/A	N/A	N/A

13. Status

Broke it as needed. Other than that, success!