Object Oriented Programming lab

Spring2025

Assignment #13

Umar Farooq
09-131242-088
BSE-2B

DEPARTMENT OF SOFTWARE ENGINEERING
BAHRIA UNIVERSITY ISLAMABAD CAMPUS

Task 01: Design a program that performs the following steps:

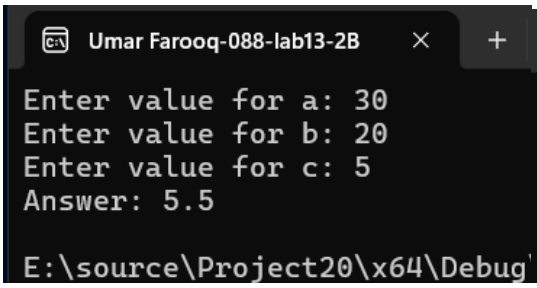
1. Start the program.
2. Declare three variables: a, b, and c.
3. Prompt the user to input values for a, b, and c.
4. Use a try block to perform the following:
    - If (a - b) != 0, calculate d = (a + b + c) / (a - b) and display the result.
    - Otherwise, throw an exception indicating that division by zero is not allowed.
5. Use a catch block to handle the exception and display an appropriate error message.
6. End the program.

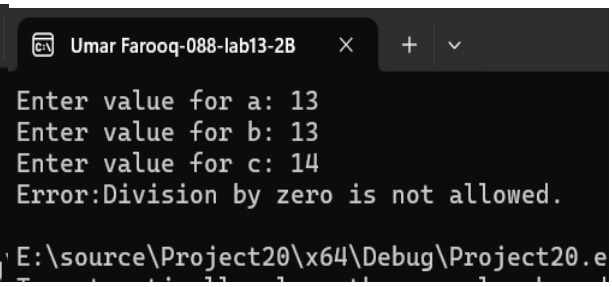
Code:

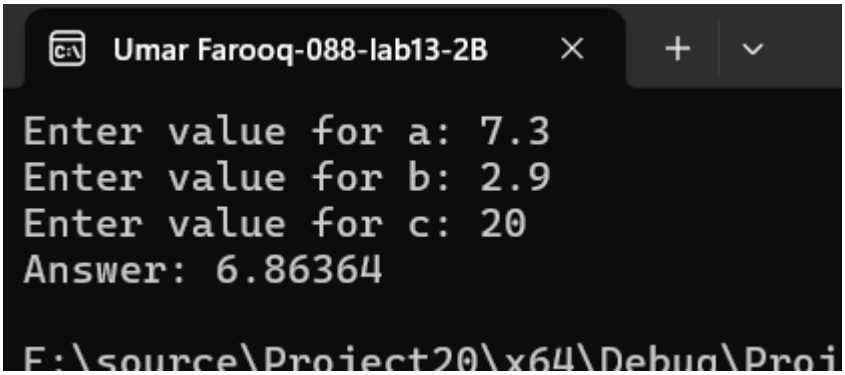
```
#include <iostream>
using namespace std;
double division(double x, double y, double z)
{
	if ((x - y) == 0)
	{
		throw "Error:Division by zero is not allowed.";
	}
	else
	{
		return (x + y + z) / (x - y);
	}
}
int main()
{
	double a, b, c;
	cout << "Enter value for a: ";
	cin >> a;
	cout << "Enter value for b: ";
	cin >> b;
	cout << "Enter value for c: ";
	cin >> c;
	double z;
	try {
		z = division(a, b, c);
		cout << "Answer: " << z << endl;
```

```
        }
        catch (const char* msg) {
                cerr << msg << endl;
        }
        return 0;
}
```

Output:







Task 02: Create a program that performs the following steps:

Step 1: Start the program.

Step 2: Declare and define the function test ().

Step 3: Within the try block, check whether the value is greater than zero or not.

a. If the value is greater than zero, throw the value and catch the corresponding exception.

b. Otherwise, throw a character and catch the corresponding exception.

Step 4: Read the integer and character values for the function test ().

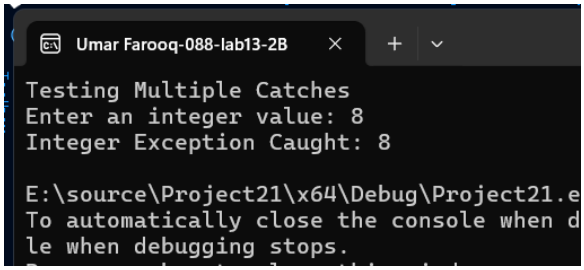
Step 5: Stop the program.

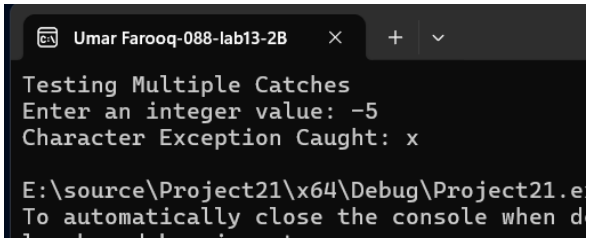
Code:

```
#include <iostream>
using namespace std;
void test(int a)
{
        try {
                if (a > 0)
```

```
                {
                        throw a;
                }
                else
                {
                        throw 'x';
                }
        }
        catch (int b)
        {
                cout << "Integer Exception Caught: " << b << endl;
        }
        catch (char b)
        {
                cout << "Character Exception Caught: " << b << endl;
        }
}
int main()
{
        int num;
        cout << "Testing Multiple Catches" << endl;
        cout << "Enter an integer value: ";
        cin >> num;
        test(num);
        return 0;
}
```

Output:





Task 03: Defining and using your own exceptions:

Write a function named **calculateAverage()** that takes four integer arguments representing the marks obtained in four different courses during a semester and returns their average as a float. The function must ensure that all the input marks are

within the valid range of 0-100, **inclusive**. If any of the marks fall outside this range, the function should throw a user-defined exception named **OutOfRangeException**. You are required to define this custom except for class yourself, as demonstrated in the lecture. In addition, provide a detailed specification for the **calculateAverage()** function in the form of comments written above the function definition. These comments should clearly describe the **preconditions** (what conditions must be true before calling the function), **postconditions** (what the function guarantees after execution), **invariants** (conditions that remain true during the execution), and the **exception(s)** that might be thrown, including the circumstances under which they are thrown. This specification will help the caller understand how the function behaves and how to use it correctly.

Code:

```
#include <iostream>
using namespace std;
class outOfRangeException {
public:
	const char* what()
	{
		return "Error: Mark is out valid range";
	}
};
float calcAverage(int n1, int n2, int n3, int n4)
{
	if (n1 < 0 || n1 > 100 || n2 < 0 || n2 > 100 || n3 < 0 || n3 > 100 || n4 < 0 || n4
> 100)
	{
		throw outOfRangeException();
	}
	return (n1 + n2 + n3 + n4) / 4.0;
}
int main()
{
	int num1, num2, num3, num4;
	cout << "Enter marks of 4 subjects: ";
	cin >> num1 >> num2 >> num3 >> num4;
	try {
		float avg = calcAverage(num1, num2, num3, num4);
		cout << "Average: " << avg << endl;
	}
```

```
        catch (outOfRangeException error)
        {
                cout << error.what() << endl;
        }
        return 0;
}
```

Output: