Objectives

After performing this lab, students shall be able to:

- Exceptional Handling

TASK 1:

Exception Handling Practice

Consider the following C++ code:

```
int numOfItems;
double unitCost;
try
{
        cout << "Enter the number of items: ";
        cin >> numOfItems;
        cout << endl;
        if (numOfItems < 0)
                throw numOfItems;
        cout << "Enter the cost of one item: ";
        cin >> unitCost;
        cout << endl;
        if (unitCost < 0)
                throw unitCost;
        cout << "Total cost: $"
                << numOfItems * unitCost << endl;
}
catch (int num)
{
        cout << "Negative number of items: " << num
                << endl;
        cout << "Number of items must be nonnegative."
                << endl;
}
catch (double dec)
{
        cout << "Negative unit cost: " << dec
                << endl;
        cout << "Unit cost must be nonnegative."
                << endl;
}
```

Answer the following:

a) What is the output if the input is 25 5.50?
b) What is the output if the input is -55 2.8?
c) What is the output if the input is 37 -4.5?
d) What is the output if the input is -10 -2.5?

TASK 2:

Define an exception **class** called **tornadoException**. The class should have two constructors including the default constructor. If the exception is thrown with the default constructor, the method **what** should return **"Tornado: Take cover immediately!"** The other constructor has a single parameter, say **m**, of the **int** type. If the exception is thrown with this constructor, the method **what** should return **"Tornado: m miles away; and approaching!"** Write a C++ driver program to test the **class tornadoException.**

TASK 3:

Write a program that prompts the user to enter a person's date of birth in numeric form such as 8-27-1980. The program then outputs the date of birth in the form: August 27, 1980. Your program must contain three exception classes: **invalidDay, invalidMonth,** and **invalidYear**. If the user enters an invalid value for day, then the program should throw and catch an **invalidDay** object. Follow similar convention for the invalid values of month. Handle leap year value with **invalidYear** exception.

TASK 4:

Write a program that create three function f1, f2 and f3.

Function f1(int,int) :

```
//print function 1 start
//f2 call
//print function 1 end
```

Function f2(int,int) :

```
//print function 2 start
//f3 call
//print function 2 end
```

Function f3(int,int) :

```
//print function 3 start
//that check 2nd number is not zero if it is zero then it throw exception "2nd Number is 0"
otherwise it will divide 1st number by 2nd Number that will throw exception a number,
//print function 3 end
```

Main:

try block

//print main start
//call f1
//print main end

catch block : that will show message
catch block : that will show integer result