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.**

Abstract Classes

TASK 3:

Consider an abstract class ***Computer*** having

- Two fields (i.e. ***companyName***, ***price***) and
- A single function named ***show()***

A class named ***Desktop*** inherits ***Computer*** class and adds fields representing

- ***color, monitor size,*** and ***processor type*** and
- Override function named ***show()*** to display values of its all attributes

A class named ***Laptop*** inherits ***Computer*** class and adds fields representing

- ***color, size, weight,*** and ***processor type*** and
- Override function named ***show()*** to display values of its all attributes

Write a ***main()*** function that instantiates objects of derived classes to access respective ***show()*** function using ***dynamic binding***.

TASK 4:

Create a class named ***Person***, which contains

- A pure virtual function named ***print()***
- Two data fields i.e. ***personName*** and ***age***

A class named ***Patient*** inherits ***Person*** class, which contains

- Two data fields i.e. ***diseaseType*** and ***recommendedMedicine***
- Overridden function ***print()*** to display all details relevant to a patient

A class named ***MedicarePatient*** inherited from class ***Patient***, which holds

- A data field representing the ***name of the hospital***
- A data filed representing the ***name of the ward***
- A data field representing ***room number***
- Overridden function ***print()*** to display all details relevant to a patient

END