## East West University

**Department of Computer Science and Engineering**

## A/2, Jahurul Islam Avenue, Jahurul Islam City, Aftabnagar, Dhaka

**Lab Manual:** 08

**Lab Topic:** Exception Handling

**Course Code:** CSE110 (Object Oriented Programming)

**Course Instructor:** Tanni Mittra, Senior Lecturer, CSE

**Lab Objective**

1. **Learn a** mechanismto handle Exception in Java program

**Lab Activities:**

1. [**Built-in Exceptions**](https://www.geeksforgeeks.org/built-exceptions-java-examples/) **Handle**

class ArithmeticException\_Demo

{

    public static void main(String args[])

    {

        try {

            int a = 30, b = 0;

            int c = a/b;

            System.out.println ("Result = " + c);

        }

        catch(ArithmeticException e) {

System.out.println (e);

            System.out.println ("Can't divide a number by 0");

        }

    }

}

Class Testthrows1 {

Void m () throws IOException

{

throw new IOException("device error");//checked exception

}

}

**Lab problem 1:**

* Write a program that creates a *Calculator* class. The class contains two variables of integer type. Design a constructor that accepts two values as parameter and set those values.
* Design four methods named *Add ()*, *Subtract ()*, *multiply ()*, *Division ( )* for performing addition, subtraction, multiplication and division of two numbers.
* For addition and subtraction, two numbers should be positive. If any negative number is entered then throw an exception in respective methods. So design an exception handler (***ArithmeticException***) in *Add ()* and *Subtract ()* methods respectivelyto check whether any number is negative or not.
* For division and multiplication two numbers should not be zero. If zero is entered for any number then throw an exception in respective methods. So design an exception handler (***ArithmeticException***) in *multiply ()* and *Division ()* methods respectivelyto check whether any number is zero or not.
* Write a main class and declare four objects of *Calculator* class. Perform addition (obj1), subtraction (obj2), multiply (obj3) and division (obj4) operations for these objects. If any non integer values are provided as input;then you should throw an exception (***NumberFormatException***) and display a message that informs the user ofthe wrong input before exiting.

**Lab problem 2:**

* Create an exception class named *MyException* that extend a base class named *Exception*
* Design a constructor in your class that accepts a string value set it to the super class constructor to display the exception message.
* Create a main class named *product*. Write a method inside the class called *productCheck(int weight)* that accepts weight of the product. Inside the method, if the weight is less than 100 then throw an exception “Product is invalid” otherwise print the weight of the product.
* Inside the main method declare single object of the product class and call the *productCheck()* method to display the weight of the product.

**Lab problem 3:**

* Write a program that meets the following requirements: Creates an array with 100 randomly chosen integers. Prompts the user to enter the index of the array, then displays the corresponding element value. If the specified index is out of bounds, display the message Outof Bounds.

**Lab problem 4:**

* Design a class named Triangle that extends GeometricObject. The class contains: Three double data fields named side1, side2, and side3 with default values 1.0 to denote three sides of the triangle. A no-arg constructor that creates a default triangle. A constructor that creates a triangle with the specified side1, side2, and side3. The accessor methods for all three data fields. A method named getArea() that returns the area of this triangle. A method named getPerimeter() that returns the perimeter of this triangle. A method named toString() that returns a string description for the triangle. The toString() method is implemented as follows: return "Triangle: side1 = " + side1 + " side2 = " + side2 +" side3 = " + side3;
* Write a test program that prompts the user to enter three sides of the triangle, a color, and a Boolean value to indicate whether the triangle is filled. The program should create a Triangle object with these sides and set the color and filled properties using the input. The program should display the area, perimeter, color, and true or false to indicate whether it is filled or not.
* In a triangle, the sum of any two sides is greater than the other side. The Triangle class must adhere to this rule. Create the IllegalTriangleException class, and modify the constructor of the Triangle class to throw an IllegalTriangleException object if a triangle is created with sides that violate the rule