Object Oriented Programming
Lab Manual 05

Exception Handling

1. Create an array of size 5 in a class named MyReader. Try to store 100 to the index 21 of the array. It should give you a runtime error. Note the first line of the error that was given.
   You do not have to handle/catch anything for this task, just note the errors.
2. In your main method, try to divide 1 by zero. Note the first line of the error that was given.
   You do not have to handle/catch anything for this task, just note the errors.
3.

```
import java.util.Scanner;
class Task3{
    public static void main(String[]args){
        Scanner sc=new Scanner(System.in);
        int x,n=sc.nextInt();
        int a[]=new int[n];

        a[5]=99;
        x=n/0;
    }
}
```

   When we enter 3 (or any number less than 6), then “a[5]=99;” line will cause error. Giving 6 or greater numbers will not cause that problem. Regardless of the number, “n/0” will always cause error. Modify the program above to handle exception ‘ArithmeticException’ and ‘ArrayIndexOutOfBoundsException’ using try, catch and finally keywords. Inside finally, just print the message “THE END”.

4. Run your solution of Task3 and give 2.5 as input. It will cause error.
   Now you see that it is hard to know every error that may occur in advance. Hence, modify your solution to Task 3 to account for all other unknown errors that can be caused from any line of the main method.

5. Change and complete the MyReader class so that you can create your own exception ‘NotIntegerException’. There will be a method int readInteger( ). This

method will take a String form the user. Then it will check if the number was ‘int’ or ‘float’ by checking presence of the decimal point in the input. It will throw an exception when a floating-point number is entered. Otherwise, it will return the integer number.

6. Write a program that uses the following method, to solve equations specified by the user.

```
static double root(double A, double B, double C)
                        throws IllegalArgumentException {
    // Returns the larger of the two roots of
    // the quadratic equation A*x*x + B*x + C = 0.
    // (Throws an exception if A == 0 or B*B-4*A*C < 0.)
  if (A == 0) {
    throw new IllegalArgumentException("A can't be zero.");
  }
  else {
    double disc = B*B - 4*A*C;
    if (disc < 0)
      throw new IllegalArgumentException("Discriminant < zero.");
    return  (-B + Math.sqrt(disc)) / (2*A);
  }
}
```

Your program should allow the user to specify values for A, B, and C. It should call the method to compute a solution of the equation. If no error occurs, it should print the root. However, if an error occurs, your program should catch that error and print an error message. After processing one equation, the program should ask whether the user wants to enter another equation. The program should continue until the user answers no..

7. Write a class TestPrimeException and add a method *boolean testPrime(int a)* in this class. This method will check if integer *a* is a prime number. It will throw “*NotPrimeException”* if a is not Prime. Now write a new exception class named ‘NotPrimeException’ that takes an integer as a parameter and when it is displayed the integer (in this case “a”) is shown. Give necessary codes.