Assignment: 6
Deadline: 7 March, 11:59 pm

Question 1: When a program fails due to an uncaught exception, the system automatically prints out the exception's stack trace. If the failure is not easily reproducible, it may be difficult or impossible to get any more information. Therefore, it is critically important that the exception's toString() method return, as much information as possible concerning the cause of the failure. In other words, the detail message of an exception should capture the failure for subsequent analysis. To capture the failure, the detail message of an exception should contain the values of all parameters and fields that "contributed to the exception."

- Create your own MyIndexOutOfBoundException Class. It should contain these parameters
    - lowerBound - the lowest legal index value.
    - upperBound - the highest legal index value.
    - index - the current index value.
- Test your code in main method, by creating an indexOutOfBoundException. Output error message should be like this:

  "Error Message: Index: 10, but Lower bound: 0, Upper bound: 9"

Question 2: Create a class called Number. Write a method called count() which can calculate quotient of two integers and use try...catch to deal with all possible exception.

Question 3: What is the problem with below program? Recommend a solution.

```
import java.io.FileNotFoundException;
import java.io.IOException;

public class TestException {

        public static void main(String[] args) {
```

```
            try {
                testExceptions();
            } catch (FileNotFoundException | IOException e) {
                e.printStackTrace();
            }
        }



        public static void testExceptions() throws IOException,
FileNotFoundException{

        }
}
```

Question 4: What is the problem with below program? Recommend a solution.

```
public class ExceptionHandling
{
   public static void main(String[] args)
   {
      try
      {
         int i = Integer.parseInt("abc");
      }

      catch(Exception ex)
      {
         System.out.println("This block handles all exception types");
      }

      catch(NumberFormatException ex)
      {

      }
   }
}
```

Question 5: How would you differentiate between throw, throws and throwable in Java?