Lab 7 –Handling Exceptions
Answer the following questions.

Instructor-led Demo:

1. Write a program that meets the following requirements:
    a. Create an array with one hundred randomly chosen integers.
    b. Cause an exception, *ArrayIndexOutOfBoundsException*, display the message "Out Of Bound". You can display all the array elements using looping.

Exercise:

1. "Passing Command-Line Arguments" is a simple command-line calculator. Note that the program terminates if any operand is non-numeric. Write a program with an exception handler that deals with non-numeric operands. Your program should display a message that informs the user of the wrong operand type before existing. For example,

Command arguments	**Output**
3 + 4	3 + 4 = 7
3/2 + 4	Wrong input: 3/2

2. Given the `Loan` class below:

```
1.  package loan;
2.
3.  import java.util.Date;
4.
5.  public class Loan {
6.
7.        private double annualInterestRate;
8.        private int numberOfYears;
9.        private double loanAmount;
10.       private java.util.Date loanDate;
11.
12.       public Loan() {
13.             // TODO Auto-generated constructor stub
14.       }
15.
16.       public Loan(double annualInterestRate, int
    numberOfYears,
17.                   double loanAmount) {
18.             super();
19.             this.annualInterestRate = annualInterestRate;
20.             this.numberOfYears = numberOfYears;
21.             this.loanAmount = loanAmount;
22.             this.loanDate = new java.util.Date();
23.       }
24.
```

```
25.       public double getAnnualInterestRate() {
26.             return annualInterestRate;
27.       }
28.
29.       public void setAnnualInterestRate(double
    annualInterestRate) {
30.             this.annualInterestRate = annualInterestRate;
31.       }
32.
33.       public int getNumberOfYears() {
34.             return numberOfYears;
35.       }
36.
37.       public void setNumberOfYears(int numberOfYears) {
38.             this.numberOfYears = numberOfYears;
39.       }
40.
41.       public double getLoanAmount() {
42.             return loanAmount;
43.       }
44.
45.       public void setLoanAmount(double loanAmount) {
46.             this.loanAmount = loanAmount;
47.       }
48.
49.       public java.util.Date getLoanDate() {
50.             return loanDate;
51.       }
52.
53.       public double monthlyPayment(){
54.             return 0.0;//return actual monthly payment
55.       }
56.
57.       public double totalPayment(){
58.             return 0.0;//return total payment
59.       }
60.
61. }
```

Modify the `Loan` class to throw `IllegalArgumentException` if the loan amount, interest rate or number of years is less than or equal to zero.

3. Consider a Calculator program, note that number 1 and number 2 were a non-numeric string, the program would report exceptions. Modify the program with an exception handler to catch `ArithmeticException` (e.g., divided by 0) and `NumberFormatException` (e.g., input is not an integer), and display the errors in a message dialog box.