OOP Lab: Experiment 7

Submitted By: Aryan Saxena
Batch: B1
SAP Id: 500082431
Roll No.: R214220274

Exercise 1: Write a program in Java to display the names and roll numbers of students. Initialize respective array variables for 10 students. Handle ArrayIndexOutOfBoundsException, so that any such problem doesn't cause illegal termination of program.

Code:

```
class StudentRecords
{
    String Name;
    int RollNo;
    StudentRecords(){};
    StudentRecords(String n, int rollnumber)
    {
        Name = n;
        RollNo = rollnumber;
    }
    public void Print()
    {
        System.out.println("Name: " + Name + "\nRoll No: " + RollNo);
    }
}



public class IndextOutofBounds
{
    public static void main(String[] args)
    {
        StudentRecords[] obj;   //Decalring Array
        obj = new StudentRecords[10];   //Assigning Size

        obj[0] = new StudentRecords("AryanSaxena",1);
        obj[1] = new StudentRecords("ChiragSingh",2);
        obj[2] = new StudentRecords("AarushiJain",3);
        obj[3] = new StudentRecords("RohitSharma",0);
        obj[4] = new StudentRecords("SarvagyaGupta",5);
        obj[5] = new StudentRecords("ManikaRajpal",6);
        obj[6] = new StudentRecords("AvinashKumar",7);
        obj[7] = new StudentRecords("AyushJuyal", 8);
        obj[8] = new StudentRecords("VeethikaEeti",9);
        obj[9] = new StudentRecords("Supandi",10);


        try
```

```
        {
            obj[10] = new StudentRecords("Batman",11);
            obj[10].Print();
        }
        catch (ArrayIndexOutOfBoundsException e)
        {
            System.out.println("Array oversized: "  + e );
            System.out.println("Try less than 10!\n");
        }
    }
}
```

Output:

```
Array oversized: java.lang.ArrayIndexOutOfBoundsException: Index 10 out of bounds for length 10
Try less than 10!

PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7>
```

Exercise 2: Create an exception class, which throws an exception if operand is nonnumeric in calculating modules. (Use command line arguments).

Code:

```
class UserException extends Exception

{
    public UserException(String s)
    {
        super(s);
    }
}

public class CustomException
{
    public static boolean isNumeric(String str)
    {
        return str != null && str.matches("[-+]?\\d*\\.?\\d+");
    }

    public static void main(String args[])
    {

        try
        {
            if(!isNumeric(args[0])||!isNumeric(args[1]))
            {
                throw new UserException("Non-Numeric Entry!");

            }
            else
            {
                int a = Integer.parseInt(args[0]);
                int b = Integer.parseInt(args[1]);
                System.out.println("Modulus of " + a + " and " + b + ": " + (a
%b));

            }
        }
        catch (UserException e)
        {
            System.out.println("Excepion Caught!");
            System.out.println(e.getMessage() + "\n");
        }
    }
}
```

Output:

```
PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7> javac CustomException.java
PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7> java CustomException  6 4
Modulus of 6 and 4: 2
PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7> java CustomException  6 -h
Excepion Caught!
Non-Numeric Entry!

PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7> java CustomException  6 jh
Excepion Caught!
Non-Numeric Entry!

PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7> java CustomException  sdf 4
Excepion Caught!
Non-Numeric Entry!

PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7>
```

Exercise 3: Write a code to create your own exception class. Create another class, inside main method prompt user to enter a number if number is less than 500 throw instances of your custom exception class.

Code:

```
import java.util.*;

class Exception500 extends Exception
{
    public Exception500(String s)
    {
        super(s);
    }
}

class Lessthan500
{
    public static void main(String[] args)
    {
        Scanner sc= new Scanner(System.in);
        System.out.print("Enter a number: ");
        int n= sc.nextInt();
        sc.close();
        try
        {
            if(n<500)
                throw new Exception500("Value less than 500");
            else
                System.out.println("No Exception found!\n");
        }
        catch (Exception500 e)
        {
            System.out.println("Exception Caught: " + e.getMessage() + "\n");
        }
    }
}
```

Output:

```
PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP> cd "f:\UPES\Academics\2nd Year\3rd Semest
va-OOP\LAB\Experiment 7\" ; if ($?) { javac Lessthan500.java } ; if ($?) { java Lessthan500 }
Enter a number: 5453
No Exception found!

PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7> cd "f:\UPES\Academics\2n
Ps Theory\sem3-Java-OOP\LAB\Experiment 7\" ; if ($?) { javac Lessthan500.java } ; if ($?) { java Lessthan500 }
Enter a number: 3
Exception Caught: Value less than 500

PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7> cd "f:\UPES\Academics\2n
Ps Theory\sem3-Java-OOP\LAB\Experiment 7\" ; if ($?) { javac Lessthan500.java } ; if ($?) { java Lessthan500 }
Enter a number: -500
Exception Caught: Value less than 500

PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7> cd "f:\UPES\Academics\2n
Ps Theory\sem3-Java-OOP\LAB\Experiment 7\" ; if ($?) { javac Lessthan500.java } ; if ($?) { java Lessthan500 }
Enter a number: 500
No Exception found!

PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7>
```

Exercise 4: You are given two integers,a and b as input, you have to compute a/b: If a and b are not bit signed integers or if is zero, exception will occur and you have to report it. Read sample Input/Output to know what to report in case of exceptions.

Code:

```
import java.util.*;

public class InputOutputException {
    public static void main(String[] args)
    {
        int a,b;
        try
        {
            Scanner sc = new Scanner(System.in);
            System.out.println("Input a & b: ");
            a= sc.nextInt();
            b= sc.nextInt();
            sc.close();
            try
            {
                System.out.println(a + "/" + b + " = " + a/b);
            }
            catch(ArithmeticException e)
            {
                System.out.println(e);
            }
        }
        catch(InputMismatchException e)
        {
            System.out.println(e);
        }

    }
}
```

Output:

```
PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP> cd "f:\UPES\Academi
va-OOP\LAB\Experiment 7\" ; if ($?) { javac InputOutputException.java } ; if ($?) { java
Input a & b:
20 2
20/2 = 10
PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7> cd
Ps Theory\sem3-Java-OOP\LAB\Experiment 7\" ; if ($?) { javac InputOutputException.java }
Input a & b:
0 1
0/1 = 0
PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7> cd
Ps Theory\sem3-Java-OOP\LAB\Experiment 7\" ; if ($?) { javac InputOutputException.java }
Input a & b:
8 0
java.lang.ArithmeticException: / by zero
PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7> cd
Ps Theory\sem3-Java-OOP\LAB\Experiment 7\" ; if ($?) { javac InputOutputException.java }
Input a & b:
g 8
java.util.InputMismatchException
PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7> cd
Ps Theory\sem3-Java-OOP\LAB\Experiment 7\" ; if ($?) { javac InputOutputException.java }
Input a & b:
0 g
java.util.InputMismatchException
PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7>
```

Exercise 5: You are required to compute the power of a number by implementing a calculator. Create a class Calc which consists of a single method long power(int, int). This method takes two integers, a and b, as parameters and finds ab. If either a or b is negative, then the method must throw an exception which says "a and b should not be negative". Also, if both a and bare zero, then the method must throw an exception which says "a and b should not be zero"

1. lang.Exception: a and b should not be negative.

Complete the function power in class *Calc* and return the appropriate result after the power operation or an appropriate exception as detailed above.

Code:

```
import java.util.*;

public class Calc
{

    public static long power(int a, int b) throws Exception
    {
        if(a<0 || b<0)
            throw new Exception("a and b should be non-negative");
        else if(a == 0 && b == 0)
            throw new Exception("a and b should not be zero.");
        else
            return (long)Math.pow(a, b);

    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a & b: ");
        int a = sc.nextInt();
        int b = sc.nextInt();
        try
        {
            long Result = power(a,b);
            System.out.println("Calculated Result: "+Result);
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        sc.close();
    }

}
```

Output:

```
va-OOP\LAB\Experiment 7\" ; if ($?) { javac Calc.java } ; if ($?) { java Calc }
Enter a & b:
4 2
Calculated Result: 16
PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7> cd "f:\UP
Ps Theory\sem3-Java-OOP\LAB\Experiment 7\" ; if ($?) { javac Calc.java } ; if ($?) { java Calc }
Enter a & b:
0 3
Calculated Result: 0
PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7> cd "f:\UP
Ps Theory\sem3-Java-OOP\LAB\Experiment 7\" ; if ($?) { javac Calc.java } ; if ($?) { java Calc }
Enter a & b:
3 0
Calculated Result: 1
PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7> cd "f:\UP
Ps Theory\sem3-Java-OOP\LAB\Experiment 7\" ; if ($?) { javac Calc.java } ; if ($?) { java Calc }
Enter a & b:
0 0
java.lang.Exception: a and b should not be zero.
PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7> cd "f:\UP
Ps Theory\sem3-Java-OOP\LAB\Experiment 7\" ; if ($?) { javac Calc.java } ; if ($?) { java Calc }
Enter a & b:
-3 5
java.lang.Exception: a and b should be non-negative
PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7> cd "f:\UP
Ps Theory\sem3-Java-OOP\LAB\Experiment 7\" ; if ($?) { javac Calc.java } ; if ($?) { java Calc }
Enter a & b:
5 -5
java.lang.Exception: a and b should be non-negative
PS F:\UPES\Academics\2nd Year\3rd Semester\OOPs Theory\sem3-Java-OOP\LAB\Experiment 7>
```