JAVA LAB 7

Meher Shrishti Nigam 20BRS1193

EXCEPTION HANDLING

1. Read the Registration Number and Mobile Number of a student. If the Registration Number does not contain exactly 9 characters or if the Mobile Number does not contain exactly 10 digits, throw an IllegalArgumentException stating that "Registration number does not contain 9 characters" and "Mobile number does contain 10 digits". If the Mobile Number contains any character other than a digit, raise a NumberFormatException. If the Registration Number contains any character other than digits and alphabets, throw a NoSuchElementException. If they are valid, print the message ‘valid’ else ‘invalid’.

CODE:

```
import java.util.regex.*;
import java.util.NoSuchElementException;

class Student_20BRS1193{
    String reg_no;
    String mobile_no;

    Student_20BRS1193(String reg_no, String mobile_no){
        validate(reg_no, mobile_no);
        System.out.println("Valid");
        this.reg_no = reg_no;
        this.mobile_no = mobile_no;
   }
    private void validate(String reg_no, String mobile_no){
        if(reg_no.length() != 9)
            throw new IllegalArgumentException("Invalid.\nRegistration
number does not contain 9 characters");
        if(mobile_no.length() != 10)
            throw new IllegalArgumentException("Invalid.\nMobile
number does not contain 10 characters");
        if(!isNumeric(mobile_no))
            throw new NumberFormatException("Invalid - Not Numeric.
\n");
        if(!isAlphaNumeric(reg_no))
            throw new NoSuchElementException("Invalid - Not
Alphanumeric. \n");
    }
    private Pattern patternNumeric = Pattern.compile("-
?\\d+(\\.\\d+)?");
```

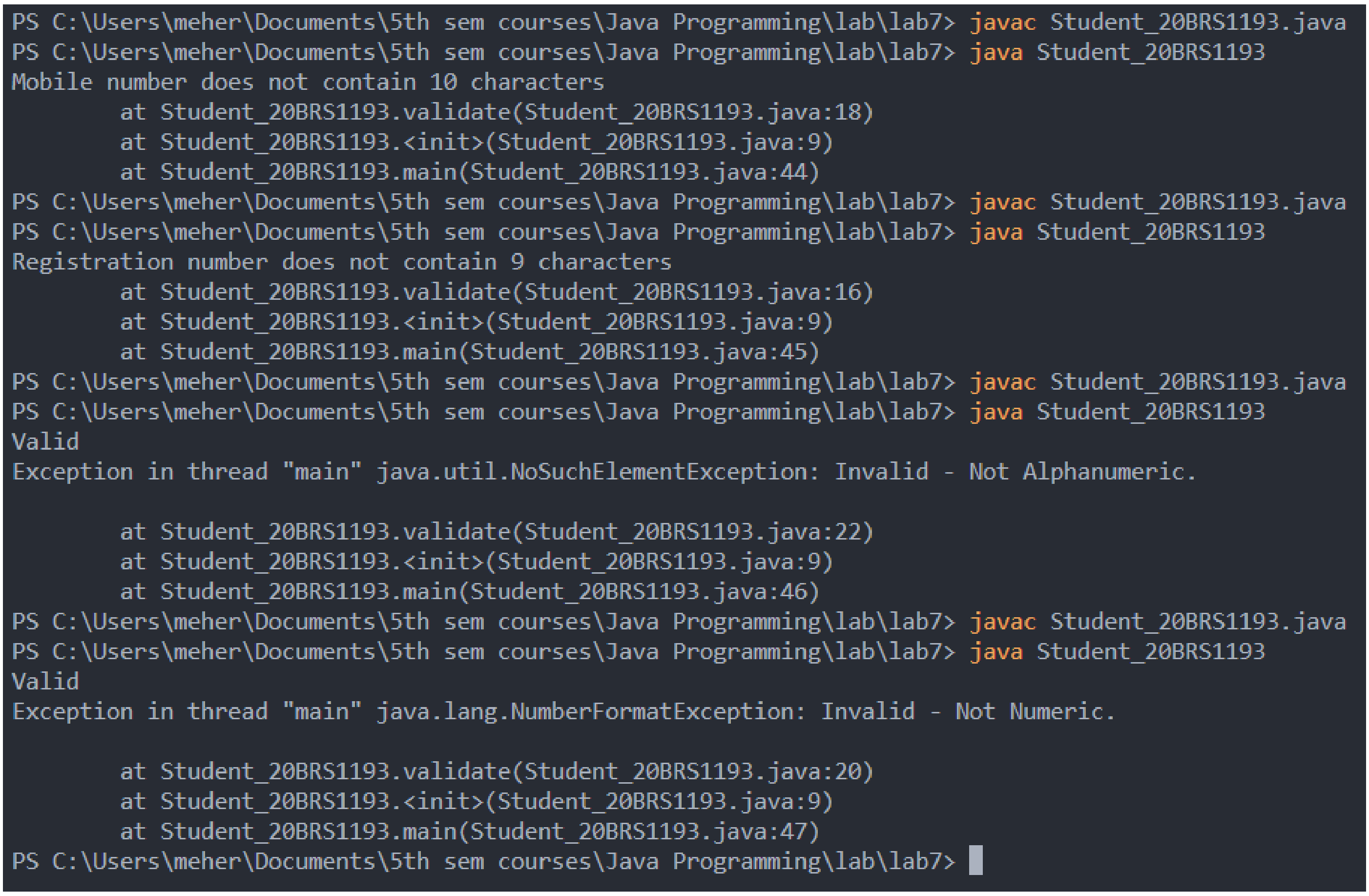
```java
    private boolean isNumeric(String strNum) {
        if (strNum == null) {
            return false;
        }
        return patternNumeric.matcher(strNum).matches();
    }

    private Pattern patternAlphaNumeric = Pattern.compile("^[a-zA-Z0-9]+$");

    private boolean isAlphaNumeric(String strNum) {
        if (strNum == null) {
            return false;
        }
        return patternAlphaNumeric.matcher(strNum).matches();
    }

    public static void main(String [] args){
        Student_20BRS1193 s1 = new Student_20BRS1193("abcd12345", "0123456789");
        Student_20BRS1193 s2 = new Student_20BRS1193("abcd12345", "012345678"); // Mobile no doesn't have 10 characters
        Student_20BRS1193 s3 = new Student_20BRS1193("abcd1234", "0123456789"); // Reg no doesn't have 9 characters
        Student_20BRS1193 s4 = new Student_20BRS1193("abcd1234_", "0123456789"); // Reg no has illegal character
        Student_20BRS1193 s5 = new Student_20BRS1193("abcd12345", "012345678w"); // Mobile no has illegal character
    }
}
```

OUTPUT:



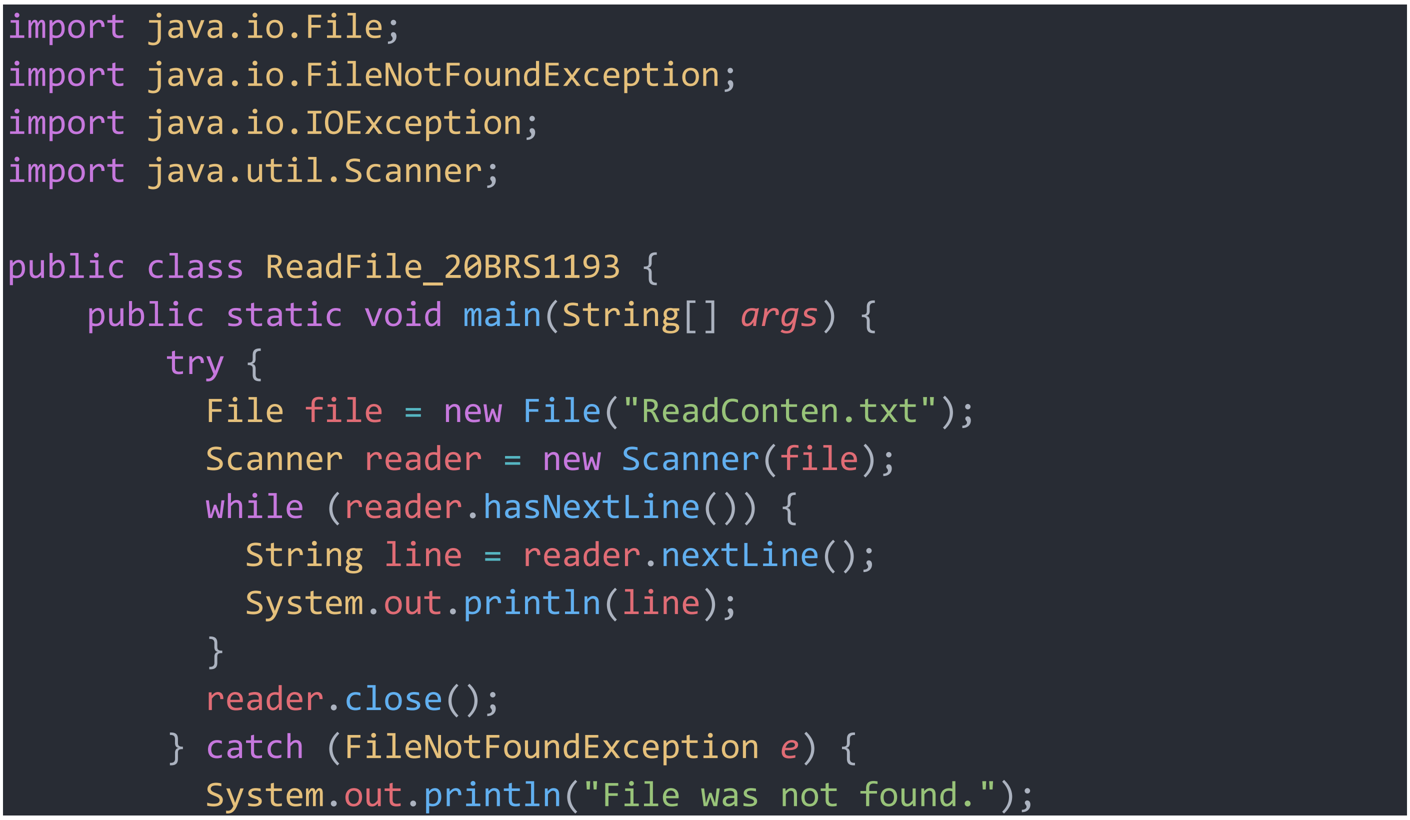
```
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> javac Student_20BRS1193.java
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> java Student_20BRS1193
Mobile number does not contain 10 characters
        at Student_20BRS1193.validate(Student_20BRS1193.java:18)
        at Student_20BRS1193.<init>(Student_20BRS1193.java:9)
        at Student_20BRS1193.main(Student_20BRS1193.java:44)
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> javac Student_20BRS1193.java
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> java Student_20BRS1193
Registration number does not contain 9 characters
        at Student_20BRS1193.validate(Student_20BRS1193.java:16)
        at Student_20BRS1193.<init>(Student_20BRS1193.java:9)
        at Student_20BRS1193.main(Student_20BRS1193.java:45)
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> javac Student_20BRS1193.java
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> java Student_20BRS1193
Valid
Exception in thread "main" java.util.NoSuchElementException: Invalid - Not Alphanumeric.

        at Student_20BRS1193.validate(Student_20BRS1193.java:22)
        at Student_20BRS1193.<init>(Student_20BRS1193.java:9)
        at Student_20BRS1193.main(Student_20BRS1193.java:46)
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> javac Student_20BRS1193.java
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> java Student_20BRS1193
Valid
Exception in thread "main" java.lang.NumberFormatException: Invalid - Not Numeric.

        at Student_20BRS1193.validate(Student_20BRS1193.java:20)
        at Student_20BRS1193.<init>(Student_20BRS1193.java:9)
        at Student_20BRS1193.main(Student_20BRS1193.java:47)
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> 
```

2. Write a java program to open a file named ReadContent.txt and read and print the contents of the file. While doing so handle all the possible exceptions (checked and unchecked) that may arise and handle them properly. Ensure that the file is closed in the finally block after all the file manipulations are carried out.

CODE:



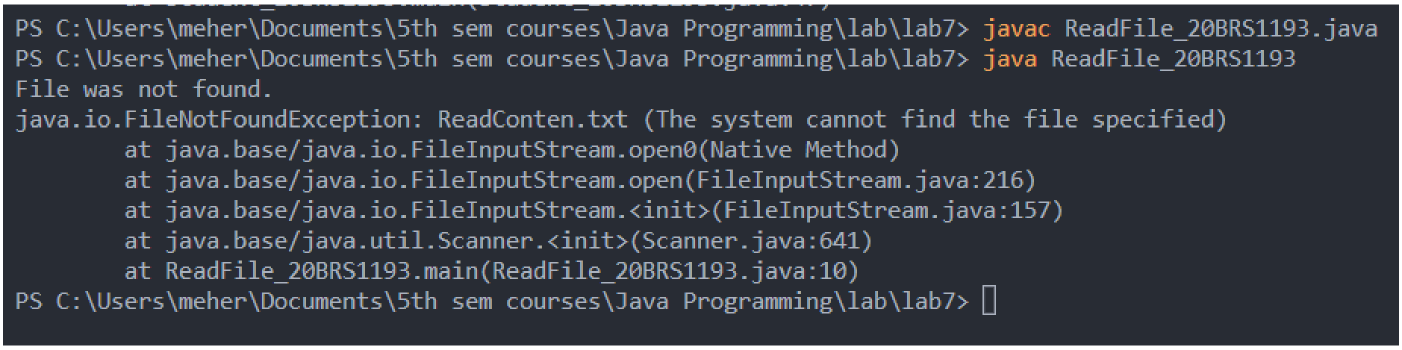
```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Scanner;

public class ReadFile_20BRS1193 {
    public static void main(String[] args) {
        try {
          File file = new File("ReadConten.txt");
          Scanner reader = new Scanner(file);
          while (reader.hasNextLine()) {
            String line = reader.nextLine();
            System.out.println(line);
          }
          reader.close();
        } catch (FileNotFoundException e) {
          System.out.println("File was not found.");
          e.printStackTrace();
        }
        catch (Exception e){
            System.out.println("Exception has occured.");
            e.printStackTrace();
        }
      }
}
```

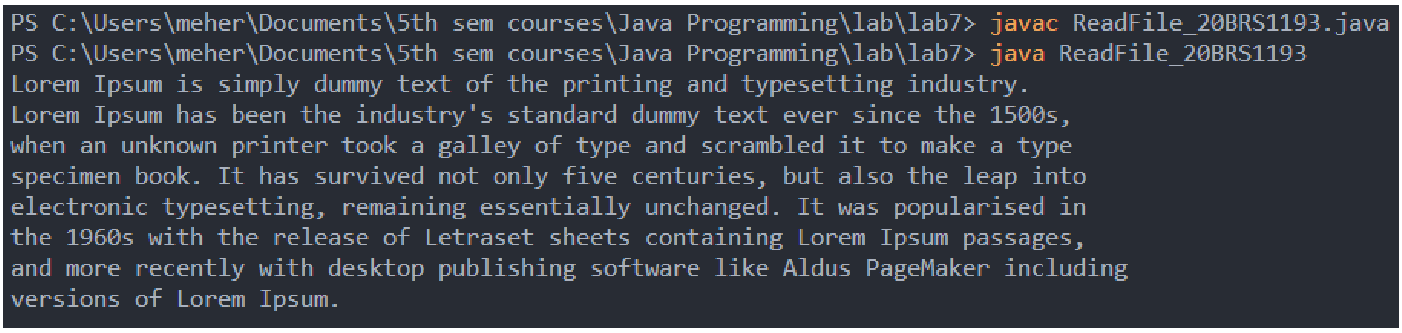
ANOTHER METHOD:

```
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
public class ReadFile2_20BRS1193 {
    public static void main(String args[]) {
        FileReader fr = null;
        try{
            File file = new File("ReadContent.txt");
            fr = new FileReader(file);
            char [] a = new char[1000];
            fr.read(a);
            for(char c : a)
                System.out.print(c);
        }
        catch (IOException e) { e.printStackTrace(); }
        finally {
            try {
                fr.close();
            }
            catch (IOException ex) {
                ex.printStackTrace();
            }
        }
    }
}
```

OUTPUT:

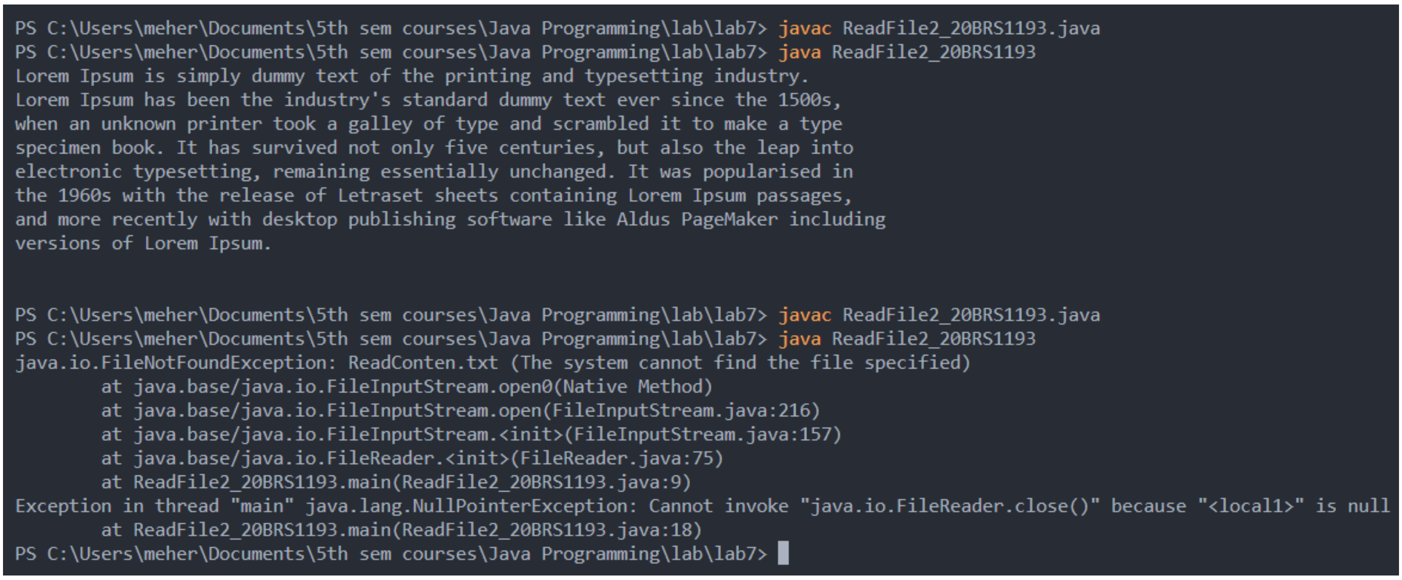


```
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> javac ReadFile_20BRS1193.java
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> java ReadFile_20BRS1193
File was not found.
java.io.FileNotFoundException: ReadConten.txt (The system cannot find the file specified)
        at java.base/java.io.FileInputStream.open0(Native Method)
        at java.base/java.io.FileInputStream.open(FileInputStream.java:216)
        at java.base/java.io.FileInputStream.<init>(FileInputStream.java:157)
        at java.base/java.util.Scanner.<init>(Scanner.java:641)
        at ReadFile_20BRS1193.main(ReadFile_20BRS1193.java:10)
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> 
```



```
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> javac ReadFile_20BRS1193.java
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> java ReadFile_20BRS1193
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
when an unknown printer took a galley of type and scrambled it to make a type
specimen book. It has survived not only five centuries, but also the leap into
electronic typesetting, remaining essentially unchanged. It was popularised in
the 1960s with the release of Letraset sheets containing Lorem Ipsum passages,
and more recently with desktop publishing software like Aldus PageMaker including
versions of Lorem Ipsum.
```

Second method –

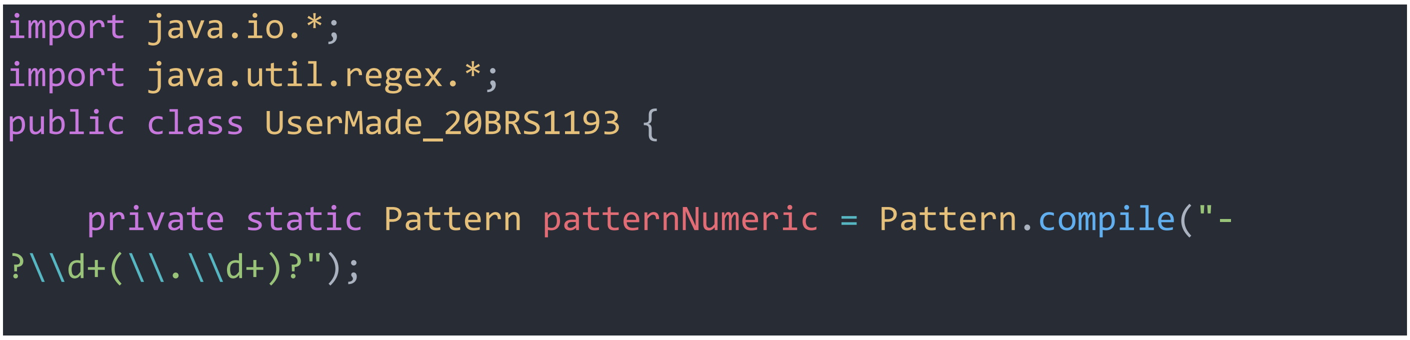


```
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> javac ReadFile2_20BRS1193.java
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> java ReadFile2_20BRS1193
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
when an unknown printer took a galley of type and scrambled it to make a type
specimen book. It has survived not only five centuries, but also the leap into
electronic typesetting, remaining essentially unchanged. It was popularised in
the 1960s with the release of Letraset sheets containing Lorem Ipsum passages,
and more recently with desktop publishing software like Aldus PageMaker including
versions of Lorem Ipsum.


PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> javac ReadFile2_20BRS1193.java
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> java ReadFile2_20BRS1193
java.io.FileNotFoundException: ReadConten.txt (The system cannot find the file specified)
        at java.base/java.io.FileInputStream.open0(Native Method)
        at java.base/java.io.FileInputStream.open(FileInputStream.java:216)
        at java.base/java.io.FileInputStream.<init>(FileInputStream.java:157)
        at java.base/java.io.FileReader.<init>(FileReader.java:75)
        at ReadFile2_20BRS1193.main(ReadFile2_20BRS1193.java:9)
Exception in thread "main" java.lang.NullPointerException: Cannot invoke "java.io.FileReader.close()" because "<local1>" is null
        at ReadFile2_20BRS1193.main(ReadFile2_20BRS1193.java:18)
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> 
```

3. Write a java program that will throw the following user defined messages when the respective unchecked exception occurs: a. "Manipulation of null value is not correct" when a null pointer exception is encountered. b. "Dividing by zero is not permitted" when a number is divided by zero. c. "Input string contains non-numeric data" when a number format exception is encountered given that an input is a string with no numeric values.

CODE:



```
import java.io.*;
import java.util.regex.*;
public class UserMade_20BRS1193 {

    private static Pattern patternNumeric = Pattern.compile("-
?\\d+(\\.\\d+)?");
```

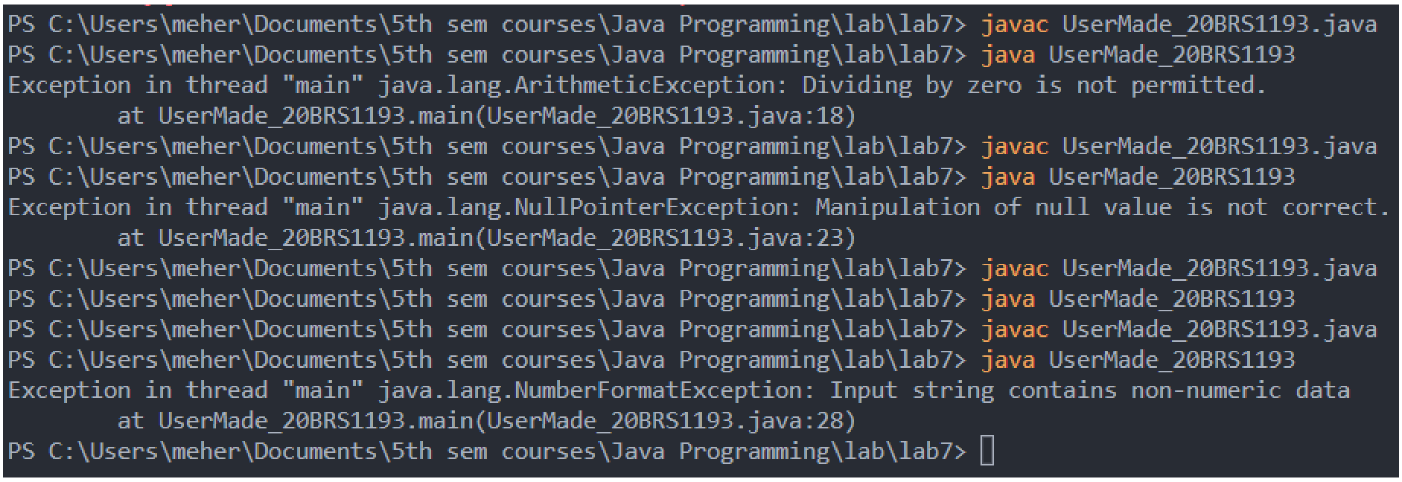
```
    private static boolean isNumeric(String strNum) {
        if (strNum == null) {
            return false;
        }
        return patternNumeric.matcher(strNum).matches();
    }
    public static void main(String [] args){

        int x = 0;
        int y = 10;
        if(x == 0)
            throw new ArithmeticException("Dividing by zero is not 
permitted.");
        int z = y/x;

        String str = null;
        if(str == null)
            throw new NullPointerException("Manipulation of null value 
is not correct.");
        System.out.println(str.length());

        String str2 = "85784702sds";
        if(!isNumeric(str2))
            throw new NumberFormatException("Input string contains 
non-numeric data");
    }
}
```

OUTPUT:



```
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> javac UserMade_20BRS1193.java
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> java UserMade_20BRS1193
Exception in thread "main" java.lang.ArithmeticException: Dividing by zero is not permitted.
        at UserMade_20BRS1193.main(UserMade_20BRS1193.java:18)
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> javac UserMade_20BRS1193.java
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> java UserMade_20BRS1193
Exception in thread "main" java.lang.NullPointerException: Manipulation of null value is not correct.
        at UserMade_20BRS1193.main(UserMade_20BRS1193.java:23)
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> javac UserMade_20BRS1193.java
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> java UserMade_20BRS1193
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> javac UserMade_20BRS1193.java
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> java UserMade_20BRS1193
Exception in thread "main" java.lang.NumberFormatException: Input string contains non-numeric data
        at UserMade_20BRS1193.main(UserMade_20BRS1193.java:28)
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> 
```

4. A swimming pool arena permits an individual to swim based on the criteria that the person is aged above 18 and knows swimming. Create two user defined exceptions namely SwimmingNotKnown and InvalidAge exceptions. Based

on the input age and swimming status entered by the user, the program should check and display the appropriate exception message

CODE:

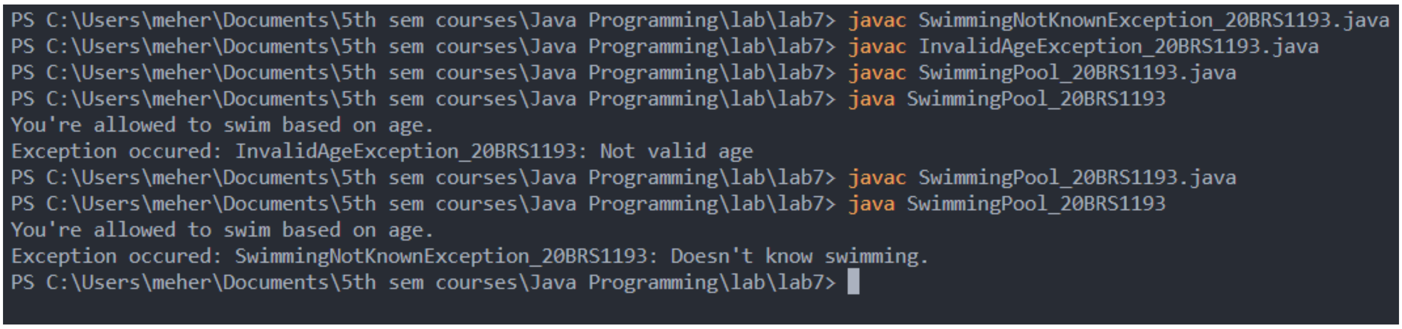
```
public class SwimmingNotKnownException_20BRS1193 extends Exception {
    SwimmingNotKnownException_20BRS1193(String s){
        super(s);
    }
}
```

```
public class InvalidAgeException_20BRS1193 extends Exception{
    InvalidAgeException_20BRS1193(String s){
        super(s);
    }
}
```

```
public class SwimmingPool_20BRS1193 {

        static void validate(int age, boolean knowsSwimming) throws
InvalidAgeException_20BRS1193, SwimmingNotKnownException_20BRS1193{
            if(age<18)
                throw new InvalidAgeException_20BRS1193("Not valid
age");
            if(!knowsSwimming)
                throw new SwimmingNotKnownException_20BRS1193("Doesn't
know swimming.");
            else
                System.out.println("You're allowed to swim based on
age.");
        }
        public static void main(String args[]){
        try{
            validate(18, true);
            validate(13, true);
            validate(34, false);
        }
        catch(Exception m){
            System.out.println("Exception occured: " + m);
        }
    }
}
```

OUTPUT:



```
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> javac SwimmingNotKnownException_20BRS1193.java
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> javac InvalidAgeException_20BRS1193.java
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> javac SwimmingPool_20BRS1193.java
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> java SwimmingPool_20BRS1193
You're allowed to swim based on age.
Exception occured: InvalidAgeException_20BRS1193: Not valid age
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> javac SwimmingPool_20BRS1193.java
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7> java SwimmingPool_20BRS1193
You're allowed to swim based on age.
Exception occured: SwimmingNotKnownException_20BRS1193: Doesn't know swimming.
PS C:\Users\meher\Documents\5th sem courses\Java Programming\lab\lab7>
```