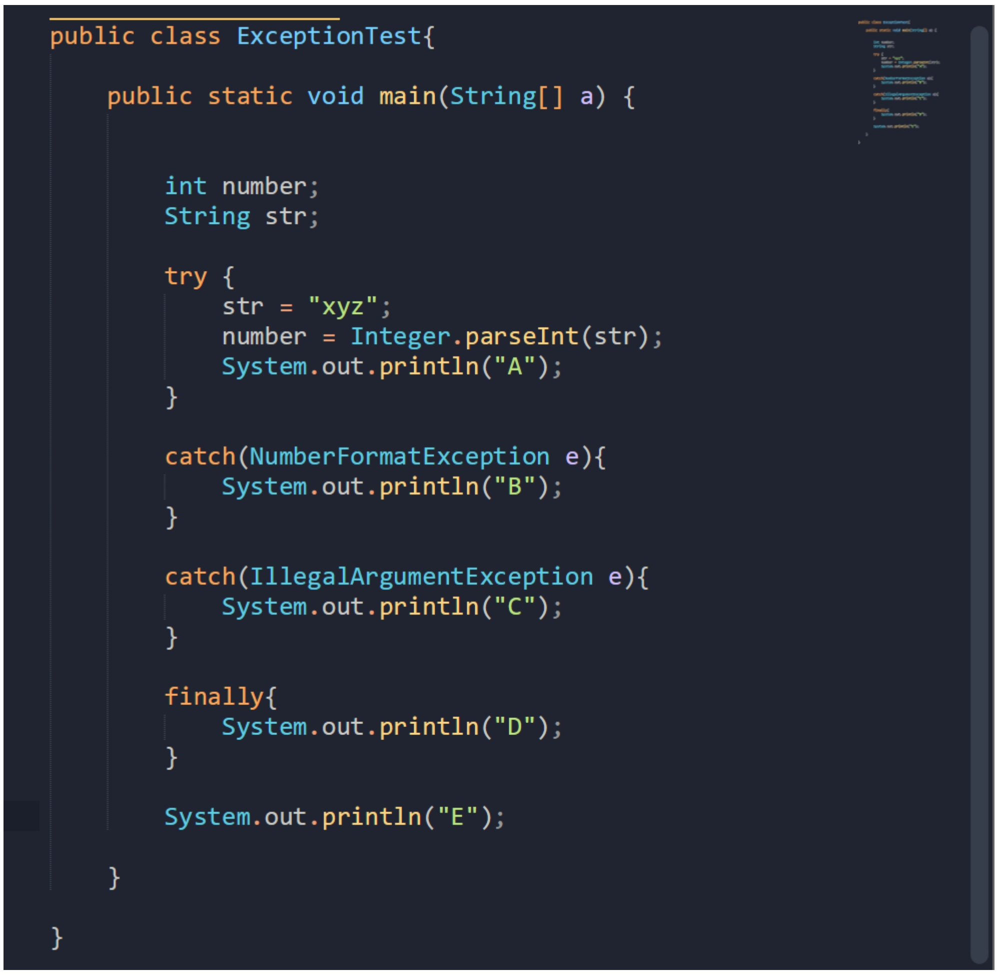
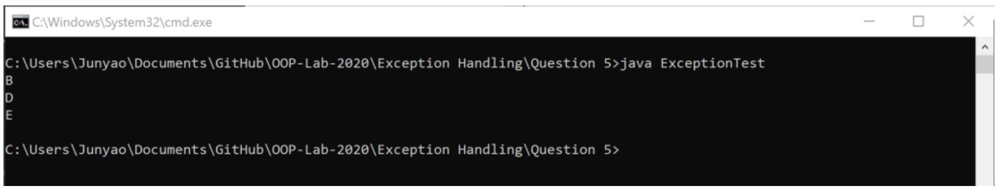
Page 278 Exercise 3 Try-Block & Catch-Block

Q5



```java
public class ExceptionTest{

    public static void main(String[] a) {

        int number;
        String str;

        try {
            str = "xyz";
            number = Integer.parseInt(str);
            System.out.println("A");
        }

        catch(NumberFormatException e){
            System.out.println("B");
        }

        catch(IllegalArgumentException e){
            System.out.println("C");
        }

        finally{
            System.out.println("D");
        }

        System.out.println("E");

    }

}
```

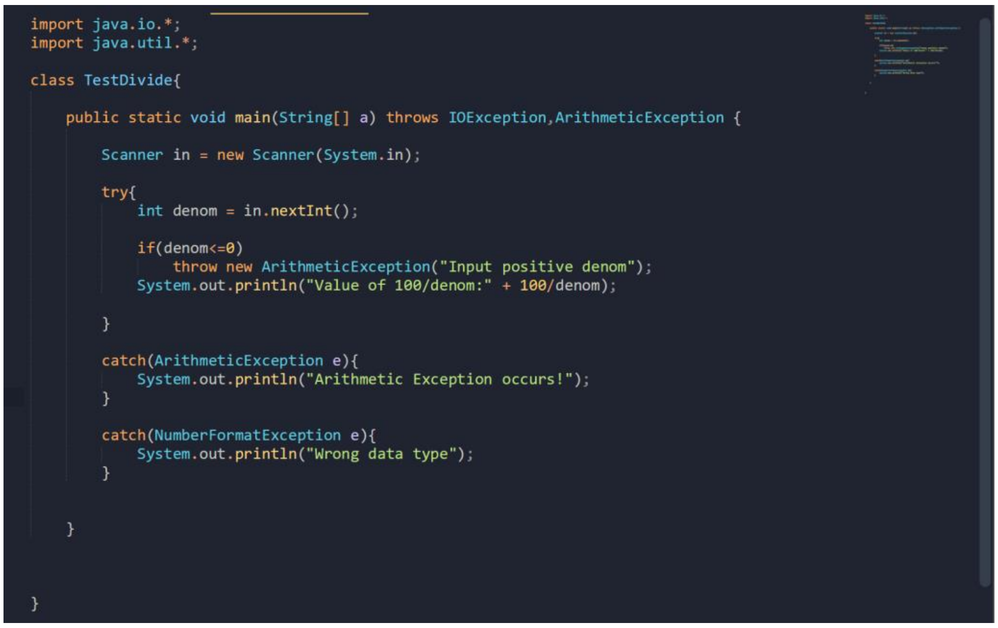


C:\Windows\System32\cmd.exe

```
C:\Users\Junyao\Documents\GitHub\OOP-Lab-2020\Exception Handling\Question 5>java ExceptionTest
B
D
E

C:\Users\Junyao\Documents\GitHub\OOP-Lab-2020\Exception Handling\Question 5>
```

B is printed as the thrown exception is instance of NumberFormatException. Then D is printed as the finally block is always executed. The last output, E is printed, as the program return to the main function.

Q6



```
import java.io.*;
import java.util.*;

class TestDivide{

    public static void main(String[] a) throws IOException,ArithmeticException {

        Scanner in = new Scanner(System.in);

        try{
            int denom = in.nextInt();

            if(denom<=0)
                throw new ArithmeticException("Input positive denom");
            System.out.println("Value of 100/denom:" + 100/denom);

        }

        catch(ArithmeticException e){
            System.out.println("Arithmetic Exception occurs!");
        }

        catch(NumberFormatException e){
            System.out.println("Wrong data type");
        }


    }


}
```

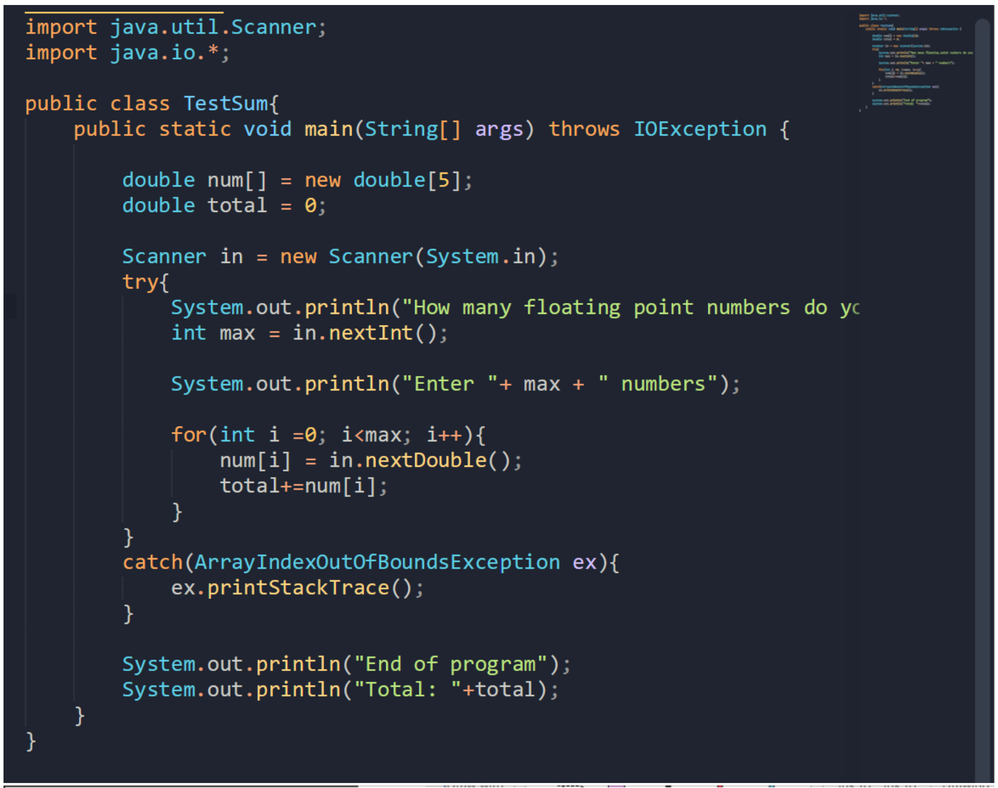
This program declared to be capable of throwing exception that is instance of IOException and ArithmeticException. The program accepts input from the user through the console input. In the try block, the program attempts to use the inputted number as a denominator to divide 100 with it. The program will catch exception thrown by the program, and display corresponding message such that: “ArithmeticException occurs!” Will be displayed if ArithmeticException occurs, and “Wrong data type” will be displayed if NumberFormatException occurred.

Page 286 Exercise 4 – Throw Exception

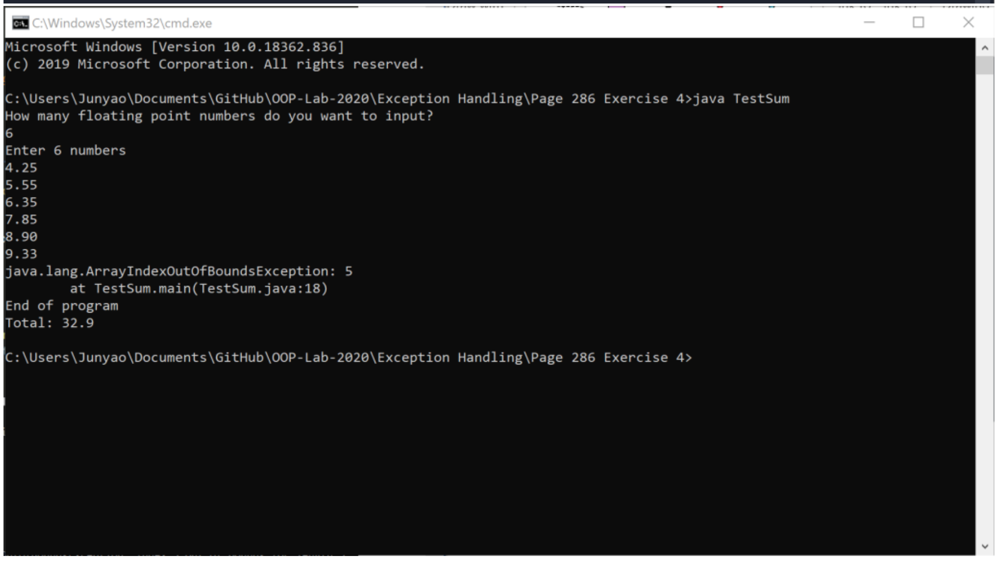
Q1

Because the exception has already been caught at the scope in which the rethrow expression occurs, it is rethrown out to the next enclosing try block. Therefore, it cannot be handled by catch blocks at the scope in which the rethrow expression occurred. Any catch blocks for the enclosing try block have an opportunity to catch the exception.

Q2



```java
import java.util.Scanner;
import java.io.*;

public class TestSum{
    public static void main(String[] args) throws IOException {

        double num[] = new double[5];
        double total = 0;

        Scanner in = new Scanner(System.in);
        try{
            System.out.println("How many floating point numbers do yo
            int max = in.nextInt();

            System.out.println("Enter "+ max + " numbers");

            for(int i =0; i<max; i++){
                num[i] = in.nextDouble();
                total+=num[i];
            }
        }
        catch(ArrayIndexOutOfBoundsException ex){
            ex.printStackTrace();
        }

        System.out.println("End of program");
        System.out.println("Total: "+total);
    }
}
```



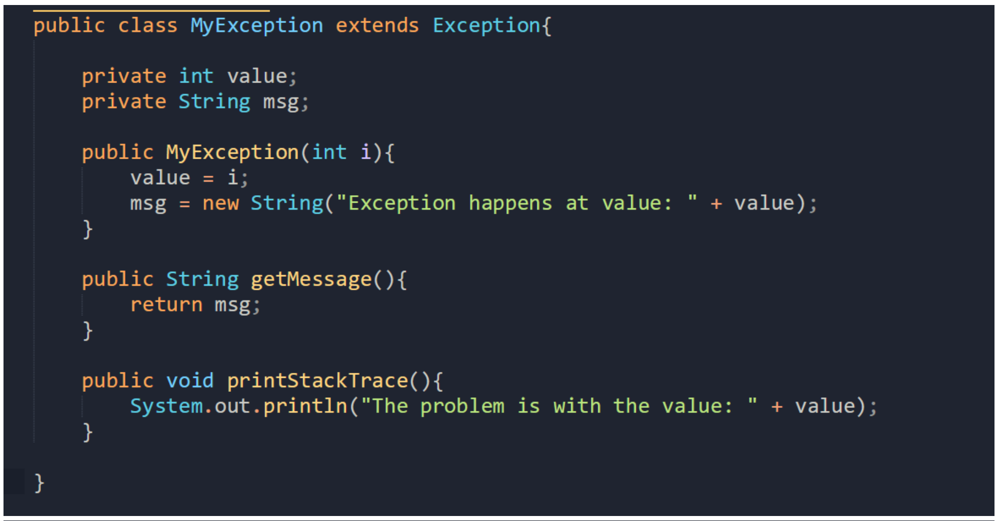
C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.18362.836]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Junyao\Documents\GitHub\OOP-Lab-2020\Exception Handling\Page 286 Exercise 4>java TestSum
How many floating point numbers do you want to input?
6
Enter 6 numbers
4.25
5.55
6.35
7.85
8.90
9.33
java.lang.ArrayIndexOutOfBoundsException: 5
        at TestSum.main(TestSum.java:18)
End of program
Total: 32.9

C:\Users\Junyao\Documents\GitHub\OOP-Lab-2020\Exception Handling\Page 286 Exercise 4>
```

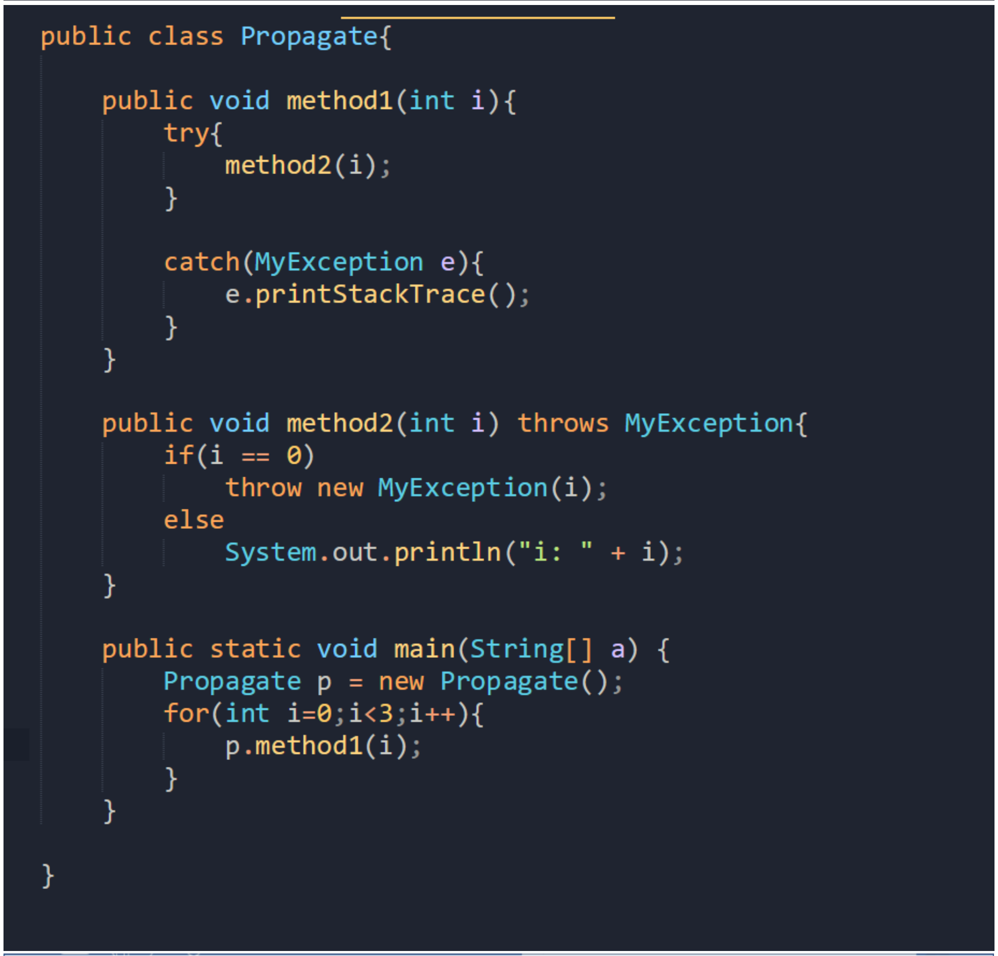
The exception, ArrayIndexOutofBoundsException is self explanatory: the declared array num[] is an array of double data, which only have a length of 5. The attempt to input 6 number failed as it has exceed the limit of the array.

Page 287 Exercise 5 – Create Own Exception

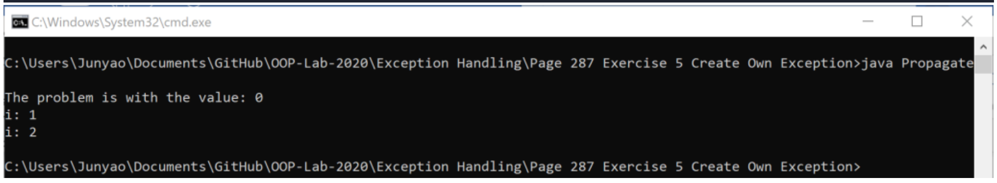
Q1



```java
public class MyException extends Exception{

    private int value;
    private String msg;

    public MyException(int i){
        value = i;
        msg = new String("Exception happens at value: " + value);
    }

    public String getMessage(){
        return msg;
    }

    public void printStackTrace(){
        System.out.println("The problem is with the value: " + value);
    }

}
```



```java
public class Propagate{

    public void method1(int i){
        try{
            method2(i);
        }

        catch(MyException e){
            e.printStackTrace();
        }
    }

    public void method2(int i) throws MyException{
        if(i == 0)
            throw new MyException(i);
        else
            System.out.println("i: " + i);
    }

    public static void main(String[] a) {
        Propagate p = new Propagate();
        for(int i=0;i<3;i++){
            p.method1(i);
        }
    }

}
```

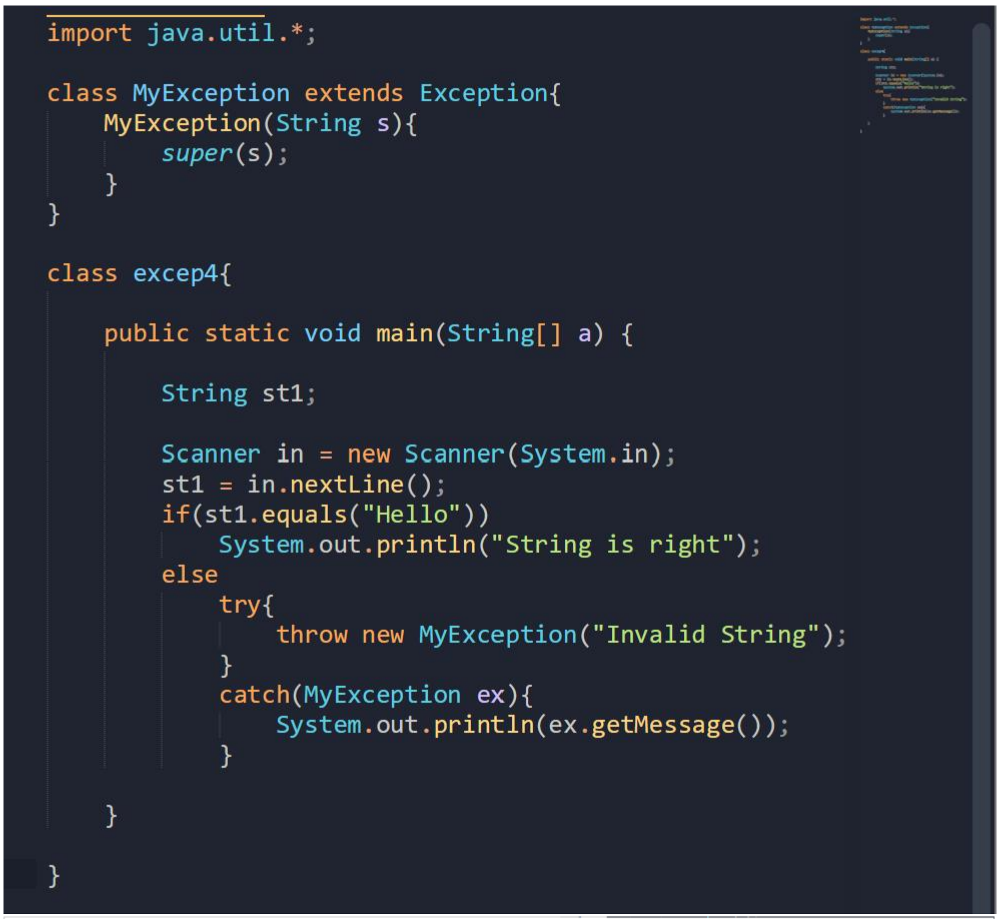


C:\Windows\System32\cmd.exe

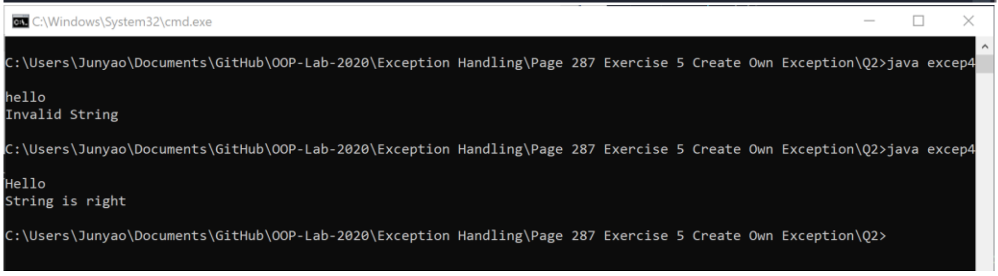
```
C:\Users\Junyao\Documents\GitHub\OOP-Lab-2020\Exception Handling\Page 287 Exercise 5 Create Own Exception>java Propagate

The problem is with the value: 0
i: 1
i: 2

C:\Users\Junyao\Documents\GitHub\OOP-Lab-2020\Exception Handling\Page 287 Exercise 5 Create Own Exception>
```

In the main program, a loop is presented, and method1 is called three times using parameter with value 0,1,2. While executing Method1, the try-block in method1 calls method2, and pass the same argument it receives. Method2 will only throw exception if the value equals to zero. Hence, the first value, 0, triggers method2 to throw exception back to method1 to catch it, subsequently printing the trace stack of the exception. The following numbers did not trigger exception in method2, hence the else block in method to executed, where the output "i: [argument]" is printed.

Q2



```java
import java.util.*;

class MyException extends Exception{
    MyException(String s){
        super(s);
    }
}

class excep4{

    public static void main(String[] a) {

        String st1;

        Scanner in = new Scanner(System.in);
        st1 = in.nextLine();
        if(st1.equals("Hello"))
            System.out.println("String is right");
        else
            try{
                throw new MyException("Invalid String");
            }
            catch(MyException ex){
                System.out.println(ex.getMessage());
            }

    }

}
```



```
C:\Users\Junyao\Documents\GitHub\OOP-Lab-2020\Exception Handling\Page 287 Exercise 5 Create Own Exception\Q2>java excep4

hello
Invalid String

C:\Users\Junyao\Documents\GitHub\OOP-Lab-2020\Exception Handling\Page 287 Exercise 5 Create Own Exception\Q2>java excep4

Hello
String is right

C:\Users\Junyao\Documents\GitHub\OOP-Lab-2020\Exception Handling\Page 287 Exercise 5 Create Own Exception\Q2>
```

Errors:

1. Class Name should start with capital letters (i.e.: MyException instead of myexception)
2. String comparison should be using StringObject.equals() method instead of using comparator "=="