



**K. J. Somaiya College of Engineering, Mumbai-77**

**Batch: B2      Roll No.: 1811094**

**Experiment / assignment / tutorial No.07**

**Grade: AA / AB / BB / BC / CC / CD / DD**

**Signature of the Staff In-charge with date**

**TITLE : User Defined Exception**

**AIM :** Write a program which accepts marks of a student (between 0 to 100) and checks whether it is within the range or not. If it is within the range then it displays “marks entered successfully”, if not then it throws the exception of user defined class “MarksOutOfRangeException”. The class should contain appropriate toString method to describe object the class with the out of range marks entered by the user.

---

**Expected OUTCOME of Experiment:**

**CO4:** Demonstrate programs on interface, exceptions, multithreading and applets.

---

**Books/ Journals/ Websites referred:**

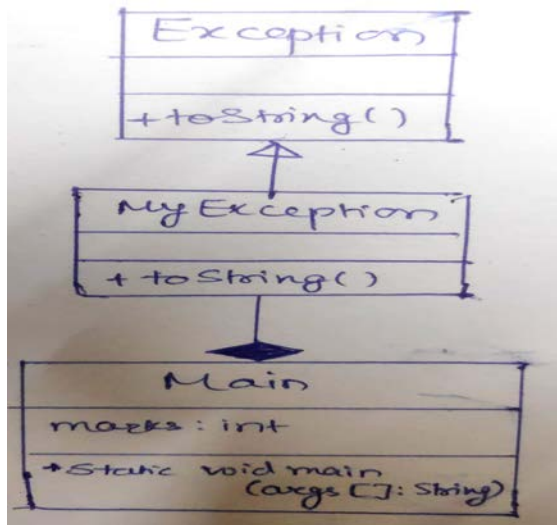
1.Ralph Bravaco , Shai Simoson , “Java Programing From the Group Up” Tata McGraw-Hill.

2.Grady Booch, Object Oriented Analysis and Design.

---

**Pre Lab/ Prior Concepts:**

**Class Diagram:**



**Algorithm:**

1. START
2. TRY
  - A. READ MARKS
  - B. IF MARKS<0 OR MARKS>100
    - a. THROW MYEXCEPTION
  - C. ELSE
    - a. PRINT "MARKS ENTERED SUCCESSFULLY"
3. CATCH
  - a. PRINT MYEXCEPTION
4. END



## K. J. Somaiya College of Engineering, Mumbai-77

### Implementation details:(printout of code)

```
import java.util.*;
import java.lang.*;
class MyException extends Exception
{
    public String toString()
    {
        return "Marks out of range";
    }
}
class Main
{
    int marks;
    public static void main(String args[]) throws Exception
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Marks");
        try
        {
            marks = s.nextInt();
            if(marks < 0 || marks > 100)
            {
                throw new MyException();
            }
            else
            {
                System.out.println("Marks entered successfully");
            }
        }
        catch (MyException e)
        {
            System.out.println(e);
        }
    }
}
```



## K. J. Somaiya College of Engineering, Mumbai-77

### Output:

```
OpenJDK Runtime Environment (build 10.0.2+13-Ubuntu-1ubuntu0.19
.04.4)
❖ javac -classpath ./run_dir/junit-4.12.jar -d . Main.java
❖ java -classpath ./run_dir/junit-4.12.jar Main
Enter Marks
87
Marks entered successfully
█

OpenJDK Runtime Environment (build 10.0.2+13-Ubuntu-1ubuntu0.19
.04.4)
❖ javac -classpath ./run_dir/junit-4.12.jar -d . Main.java
❖ java -classpath ./run_dir/junit-4.12.jar Main
Enter Marks
123
Marks out of range
█
```

### Conclusion:

The concept of try-catch block was used in the program.  
The program displays the required message if an exception is occurred.

Date: \_\_\_\_\_

Signature of faculty in-charge

### Post Lab Descriptive Questions

1. Compare throw and throws.

Throw	Throws
throw keyword is used to throw an exception explicitly.	throws keyword is used to declare one or more exceptions, separated by commas.
Only single exception is thrown by using throw.	Multiple exceptions can be thrown by using throws.
throw keyword is used within the method.	throws keyword is used with the method signature.
Syntax wise throw keyword is followed by the instance variable.	Syntax wise throws keyword is followed by exception class names.



## K. J. Somaiya College of Engineering, Mumbai-77

Checked exception cannot be propagated using throw only. Unchecked exception can be propagated using throw.

For the propagation checked exception must use throws keyword followed by specific exception class name.

2. Explain how to create a user define exception and explicitly throwing exception in program with simple example.

```
import java.util.*;
import java.lang.*;
class UserDefinedException extends Exception
{
    public String toString()
    {
        return "Cannot divide by zero\nThis is an User Defined Exception";
    }
}

class a
{
    public static void main(String args[])
    {
        Scanner s = new Scanner(System.in);
        int aa,b;
        aa = s.nextInt();
        b = s.nextInt();
        try
        {
            if(b==0)
            {
                throw new UserDefinedException();
            }
            else
            {
                System.out.println(aa/b);
            }
        }
        catch(UserDefinedException e)
        {
            System.out.println("Exception: " + e);
        }
    }
}
```

To create a user defined exception, we have to create a class which inherits Exception class.



## K. J. Somaiya College of Engineering, Mumbai-77

The default toString() method is overridden with the message we want to display if any exception occurs. In the main function, two integer inputs are taken from the user. If the second number is 0, the try block throws an exception else it prints the required result. If the exception occurs then the try block throws the exception by using 'throw' keyword. An object of the user defined Exception is created just after the 'throw' keyword. The throw keyword sends the control to the catch block which prints the required message.

3. Suppose the statement2 causes an exception in following try-catch block:

```
try {
    statement1;
    statement2;
    statement3;
}
catch(Exception1 e1) {
}
catch(Exception2 e2){
}

statement4;
```

Answer the following questions:

- Will statement3 be executed?  
Answer: NO
- If the exception is not caught, will statement4 be executed?  
Answer: NO
- If the exception is caught in the catch block, will statement4 be executed?  
Answer: YES
- If the exception is passed to the caller, will the statement4 be executed?  
Answer: NO



## **K. J. Somaiya College of Engineering, Mumbai-77**