

## CREATING OWN EXCEPTIONS

### AIM:

To write a java program to implement user defined exception handling.

### ALGORITHM:

1. Import the java packages.
2. Create a subclass of Exception named as MyException it has only a constructor plus an overloaded toString ( ) method that displays the value of the exception.
3. The exception is thrown when compute ( ) integer parameter is greater than 10.
4. The main ( ) method sets up an exception handler for MyException, then calls compute ( ) with a legal value (less than 10) and an illegal one to show both paths through the code.

### PROGRAM:

*//File Name should be **UserException.java***

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

class MyException extends Exception
{
    private int d;
    MyException(int a)
    {    d =
a;
    }

    public String toString()
    {
        return "MyException [" + d + "];"
    }
}

class UserException
{
    static void compute(int a) throws MyException
```

```
{
    System.out.println ("Called Compute(" + a + ")");
    if(a>10)
        throw new
MyException(a);
    System.out.println ("Normal Exit");
}
public static void main(String args[])
{
try
{
    compute(1);
    compute(20);
}
catch(MyException e)
{
    System.out.println("Caught " + e);
}
}
}
```

#### **NOTE:**

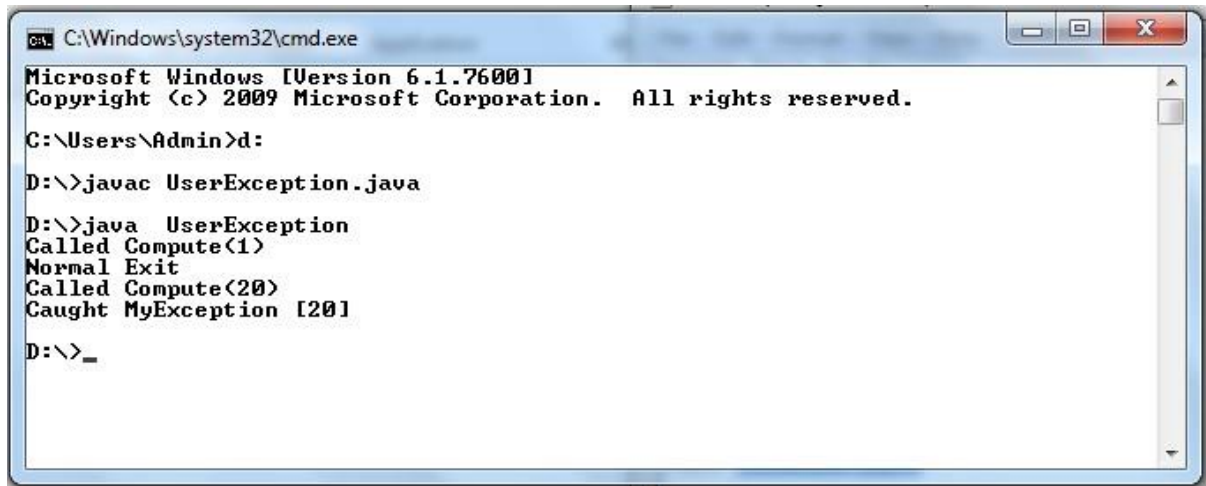
To Compile:

*javac UserException.java* To

Run:

*java UserException*

#### **OUTPUT:**

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe'. The window contains the following text:

```
Microsoft Windows [Version 6.1.7600]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
  
C:\Users\Admin>d:  
  
D:\>javac UserException.java  
  
D:\>java UserException  
Called Compute<1>  
Normal Exit  
Called Compute<20>  
Caught MyException [20]  
  
D:\>_
```

## RESULT:

Thus the Implementation for user defined exception handling has been successfully executed.