ASSIGNMENT 7 : LAB 6

Question 1

Create a file numbers.txt as follows:

```
100
9
-89
76
999999
0
-0.1
56
```

(a) Create a class *MyFileManager* having the main() loop. Using the BufferedReader class and the try-catch-finally block, read the contents of *numbers.txt*.

(b) Inject an exception by giving the name of a non-existent file e.g. *foo.txt*. Remember to release resources in the *finally*block.

(c) Write a user-defined exception - call it NegativeNumberException - to reject negative numbers while reading a file.
Create a class *MyValidator* that validates input. [*Hint: validateNumbers(...) throws NegativeNumberException*] The exception handling should be such that the programme continues reading the file even after encountering a negative number.

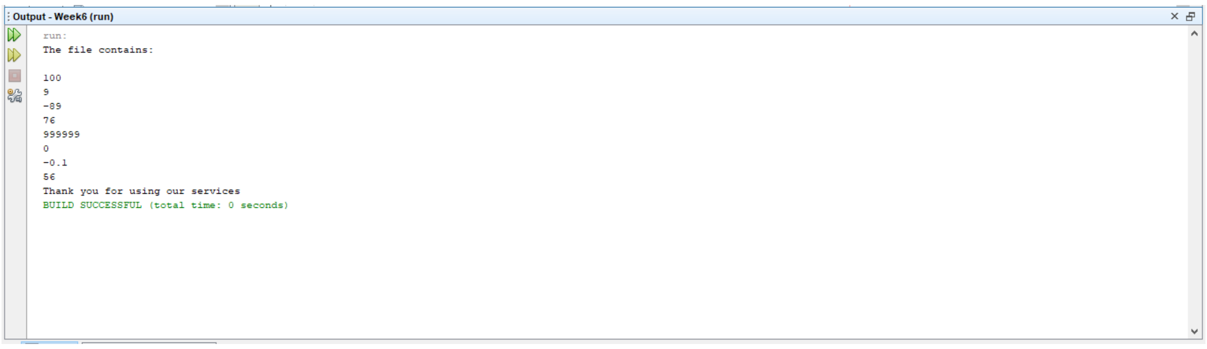
SOURCE CODE :

A)

```
/*
 * @author HARSH
 */
import java.io.*;
public class MyFileManager {



    public static void main(String args[])
    {
```

```
    try {


     FileReader myFileReader=new FileReader("C:\\Users\\bijay\\Documents\\Week6File.txt");
     BufferedReader myReader=new BufferedReader(myFileReader); //creating buffer object
     FileWriter myWriter;
     String inputLine;
     System.out.println("The file contains:");
     inputLine = myReader.readLine(); // read in a line
       while(inputLine!=null){
         System.out.println(inputLine);
         inputLine = myReader.readLine();// read in a line
     }
     myReader.close();
    }
     catch(FileNotFoundException e){
         System.out.println("Error opening the input file!  " +e.getMessage());
     }
    catch(IOException e) {
         System.out.println("IO Error!" + e.getMessage());
                 }
                finally
            {
                 System.out.println("Thank you for using our services");
            }


}
}
```

OUTPUT :

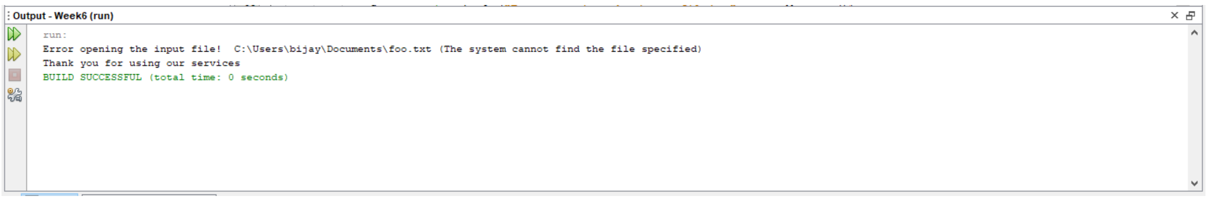


```
run:
The file contains:

100
9
-89
76
999999
0
-0.1
56
Thank you for using our services
BUILD SUCCESSFUL (total time: 0 seconds)
```

B)

```
import java.io.*;
public class MyFileManager {


   public static void main(String args[])
   {
      try {


        FileReader myFileReader=new FileReader("C:\\Users\\bijay\\Documents\\foo.txt");
        BufferedReader myReader=new BufferedReader(myFileReader); //creating buffer object
        FileWriter myWriter;
        String inputLine;
        System.out.println("The file contains:");
        inputLine = myReader.readLine(); // read in a line
          while(inputLine!=null){
            System.out.println(inputLine);
            inputLine = myReader.readLine();// read in a line
        }
      myReader.close();
      }
        catch(FileNotFoundException e){
            System.out.println("Error opening the input file!  " +e.getMessage());
        }
        catch(IOException e) {
            System.out.println("IO Error!" + e.getMessage());
                    }
        finally
                {
                    System.out.println("Thank you for using our services");
                }


}
}
```

OUTPUT



Output - Week6 (run)

```
run:
Error opening the input file!  C:\Users\bijay\Documents\foo.txt (The system cannot find the file specified)
Thank you for using our services
BUILD SUCCESSFUL (total time: 0 seconds)
```

C)

MyFileManager.java

```
import java.io.*;
public class MyFileManager {


    public static void main(String args[])throws NegativeNumbersException
    {
       try {


         FileReader myFileReader=new FileReader("C:\\Users\\bijay\\Documents\\Week6File.txt");
         BufferedReader myReader=new BufferedReader(myFileReader); //creating buffer object
         FileWriter myWriter;
         String inputLine;
         System.out.println("The file contains:");
         inputLine = myReader.readLine(); // read in a line
           while(inputLine!=null){
              double k=Double.parseDouble(inputLine);//Converting string to double
               MyValidator v=new MyValidator();
               v.validateNumbers(k);
            //System.out.println(inputLine);
            inputLine = myReader.readLine();// read in a line
        }
        myReader.close();
        }
         catch(FileNotFoundException e){
            System.out.println("Error opening the input file!  " +e.getMessage());
         }
        catch(IOException e) {
            System.out.println("IO Error!" + e.getMessage());
                }
        finally
                {
                    System.out.println("Thank you for using our services");
                }


}
}
```

MyValidator.java

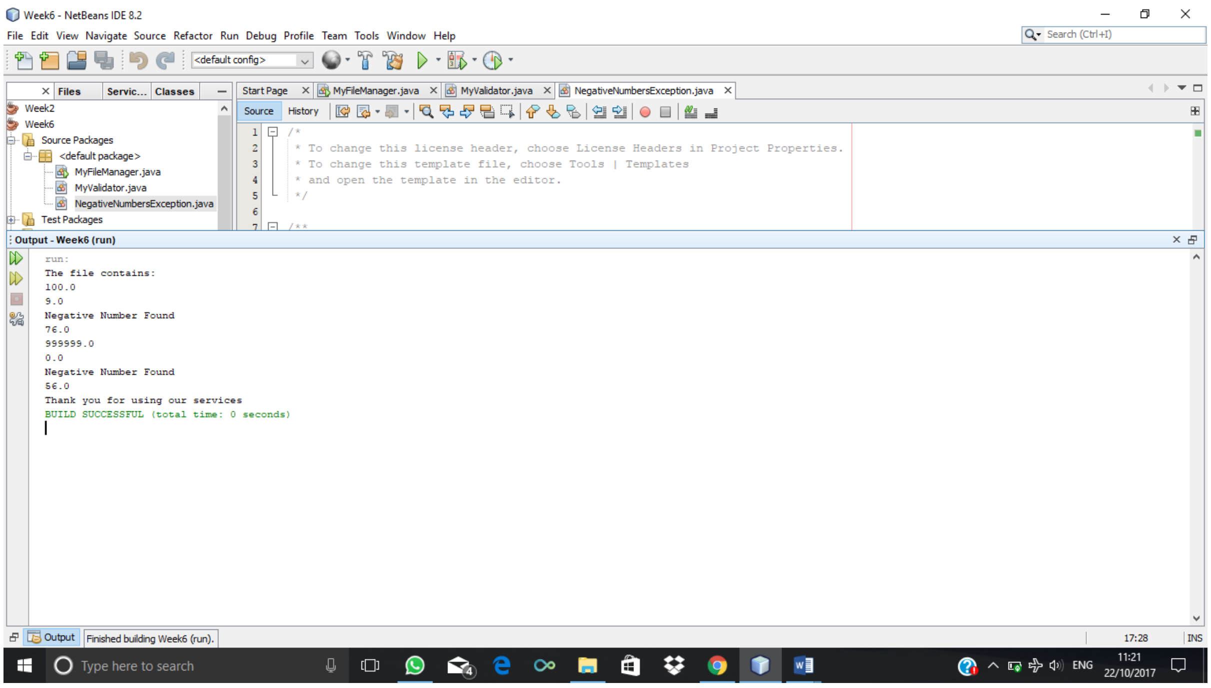
```
public class MyValidator {
    public void validateNumbers(double n)throws NegativeNumbersException {
try
{
if(n>=0) System.out.println(n);
if(n<0)
throw new NegativeNumbersException();
} catch(NegativeNumbersException e)
{
```

```
System.out.println(e.display());
}
}
}
```

NegativeNumbersException.java

```
public class NegativeNumbersException extends Exception {
    public NegativeNumbersException()// default constructor
    {
    }
 public String display()
    {
    return "Negative Number Found";
    }
}
```

OUTPUT:



```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
```

```
run:
The file contains:
100.0
9.0
Negative Number Found
76.0
999999.0
0.0
Negative Number Found
56.0
Thank you for using our services
BUILD SUCCESSFUL (total time: 0 seconds)
```

Question 2 - Triangle Class, Take 3
Refer to the javadocs for the java.lang.Comparable interface.

In Lab #5 Question 1, we focused on the return value, i.e. "a negative integer, zero, or a positive integer as this object is less than, equal to, or greater than the specified object."

Now, provide the full implementation of the specification by throwing the NullPointerException and the ClassCastException.

Why did the compiler not give an error for Lab #5 Question 1 when we did not implement exception handling?

SOURCE CODE :

ShapeManager.java

```
/**
 *
 * @author bijay
 */
import java.util.Scanner;
public class ShapeManager
{
   public static void main(String[] args)
   {
       int flag=0,cont=1;
       Scanner sc=new Scanner(System.in);
       while(cont==1)
       {
           System.out.println("Enter base of first triangle");
           double b1=sc.nextDouble();
           System.out.println("Enter height of first triangle");
           double h1=sc.nextDouble();
           System.out.println("Enter base of second triangle");
           double b2=sc.nextDouble();
           System.out.println("Enter height of second triangle");
           double h2=sc.nextDouble();
           try
```

```
        {
            if(b1==0 || h1==0 || b2==0 || h2==0)
                throw new NullPointerException();
            Triangle t1=new Triangle(b1,h1);//Creates first Triangle object
            t1.show();
            Triangle t2=new Triangle(b2,h2);//Creates second Triangle object
            t2.show();
            int cmp=t2.compareTo(t1);//Calls compareTo function
            if(cmp==0)
                System.out.println("Both their areas are equal");
            else if(cmp==1)
                System.out.println("Area of 1st Triangle is larger");
            else
                System.out.println("Area of 2nd Triangle is larger");
        }
        catch(NullPointerException e)
        {
            System.out.println("Null Pointer Exception. Error-base / height cannot be 0 or negative ");
        }
        catch(ClassCastException e)
        {
            System.out.println("Cannot convert an object to a subclass of which it is not an
instance"+e.getMessage());
        }
        System.out.println("Do you want to compare again? Press 1 if yes, else 0");
        cont=sc.nextInt();
        if(cont==1)
            flag=0;
      }
   }
}
```

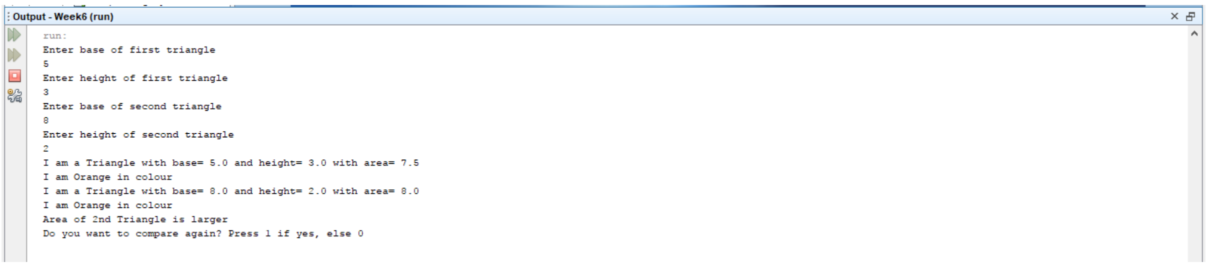
Shape.java

```
public abstract class Shape {

  //abstract class
  private String colour="Orange";
  public abstract void show();
  public abstract double getArea();
  public String getcolour()
  {
      return colour;
  }
}
```

Triangle.java

```
import java.lang.Comparable;
public class Triangle extends Shape implements Comparable<Triangle>
{
  private double base,height;
  private Triangle()//default constructor
  {
      base=0;
      height=0;
  }
  public Triangle(double base,double height)//parameterised constructor
  {
      this.base=base;
      this.height=height;
  }
  public double getArea()
  {
      return 0.5*base*height;
  }
  public void show()
  {
      System.out.println("I am a Triangle with base= "+base+" and height= "+height+" with area= "+getArea());
      System.out.println("I am "+super.getcolour()+" in colour");
  }
  public int compareTo(Triangle t)//Compares the areas of triangles
  {
      if(this.getArea()<t.getArea())
          return 1;
      else if(this.getArea()>t.getArea())
          return -1;
      else
          return 0;
  }
}
```

OUTPUT



```
run:
Enter base of first triangle
5
Enter height of first triangle
3
Enter base of second triangle
8
Enter height of second triangle
2
I am a Triangle with base= 5.0 and height= 3.0 with area= 7.5
I am Orange in colour
I am a Triangle with base= 8.0 and height= 2.0 with area= 8.0
I am Orange in colour
Area of 2nd Triangle is larger
Do you want to compare again? Press 1 if yes, else 0
```

Q. Why did the compiler not give an error for Lab #5 Question 1 when we did not implement exception handling?
Answer: In the previous question, we were passing a Triangle object as a Shape object. Since Triangle class inherits Shape class i.e Triangle “is a” Shape, there is no ClassCastException. Had it been the other way round, the compiler would show an error message because Shape is not a Triangle.