ASSIGNMENT 3

Q.1 Write a Java program to write a string in a file using FileOutputStream class.

Program:

```
import java.io.*;

public class FileO {


   public static void main(String[] args) {

    try{

       FileOutputStream fout = new FileOutputStream("abc.txt");

       String s1= "hello my name is imran";

//convert string into bytes

       byte b[]=s1.getBytes();

       fout.write(b);

       fout.close();

System.out.println("String has been written in the file !");

   }

    catch(IOException  e) {

       System.out.print(e);

         }

     }

 }
```

Output:

```
String has been written in the file !
```

Q.2 Write a Java program to read content of a file and print it on screen using FileInputStream class.

Program:

```
import java.io.*;
public class FILEI {
   public static void main(String[] args) {
    try{
      FileOutputStream fout= new FileOutputStream("test.txt");
      String msg="hello this is file stream program !";
      fout.write(msg.getBytes());
      FileInputStream fin = new FileInputStream("test.txt");
      int i=0;
      while((i=fin.read())!=-1){
         System.out.print((char)i);
      }
      fin.close();
   }
   catch(IOException e){
      System.out.print(e);
   }
   }    }
```

Output:

```
hello this is file stream program !
```

Q 3. Write a Java program which takes input from a file and write it into another file.

Program:

```
import java.io.*;
class CopyFile{

public static void main(String args[]){

   try{

              FileInputStream fin = new FileInputStream("FirstFile.txt");
              FileOutputStream fout = new FileOutputStream("secondFile.txt");

              int i=0;
              while((i=fin.read())!=-1){
                fout.write(i);

              }
              System.out.println("data is sucessfully copied !");
              FileInputStream fin1 = new FileInputStream("secondFile.txt");
              int j=0;
              while( (j= fin1.read())!=-1){
                        System.out.print((char)j);

              }

        }

        catch(IOException e){

          System.out.print(e);
        }

        }

}
```

Output:

```
data is sucessfully copied !

hello this is first file content.
```

Q.4 Write a Java program to show uses of try-catch in exception handling.

Program:

```
import java.util.Scanner;

class  Error{

public static void main(String args[]){

        int a,b,c;

        Scanner obj = new Scanner(System.in);

        System.out.print("enter value of a,b and c:");

        a = obj.nextInt();

        b = obj.nextInt();

        c = obj.nextInt();

        int x,y;

        try  {

         x = a/(b-c);   // Execption here

         System.out.println("result is : "+x);

                }

        catch(ArithmeticException e)  {

                        System.out.println("Zero division error !");   }  }

}
```

output:

```
enter value of a:5

enter value of b:2

enter value of c:2

Zero division error !
```

Q.5 Write a Java program to demonstrate throwing user defined exceptions as well as finally in exception handling.

Program:

```
import java.lang.Exception;

class MyException extends Exception{

        MyException(String msg){

        super(msg);   }

}

class TestException{

public static void main(String args[]){

        int x=5,y=1000;

        try {

                float z=(float)x/(float)y;

                if(z<0.01)    {

                throw new MyException("number is too small"); }

           }

        catch(MyException e)

        {    System.out.println("caught my exception");

           System.out.println(e.getMessage());  }

      finally  {     System.out.println("i am always here !");

        }     }   }
```

Output:

```
caught my exception

number is too small

i am always here !
```

Q.6 Write a program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 and Num2 were not integers, the program would throw a Number Format Exception. If Num2 were zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box .

Program:

```
public class NewJFrame extends javax.swing.JFrame {

   public NewJFrame() { initComponents();   }

   private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

      try{

         int x=Integer.parseInt(jTextField1.getText());

         int y=Integer.parseInt(jTextField2.getText());

         int divide=x/y;

         jTextField3.setText(" "+divide);

         if (y == 0) {

throw new ArithmeticException("Division by zero");

         }   }

      catch(NumberFormatException e)   {

jOptionPane1.showMessageDialog(null, "only Integer value is allowed.", "Error",
jOptionPane1.ERROR_MESSAGE);  }

      catch(ArithmeticException e)   {

 jOptionPane1.showMessageDialog(null, "Division by zero is notallowed.", "Error",
jOptionPane1.ERROR_MESSAGE);  }  }

   public static void main(String args[]) {

      java.awt.EventQueue.invokeLater(new Runnable() {

         public void run() {

            new NewJFrame().setVisible(true);   }

      }  );

 }
```

```
    private javax.swing.JButton jButton1;
    private javax.swing.JDialog jDialog1;
    private javax.swing.JFrame jFrame1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JOptionPane jOptionPane1;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JTextField jTextField3;
    private java.awt.Panel panel1;
}
```

output: