JAVA PROGRAMMING

LAB MANUAL

B.Tech IIYR IV SEM

OBJECTIVES:

- To teach the students basics of JAVA programs and its execution.
- To teach the students the differences between C++ and Java programming.
- To make the students learn concepts like packages and interfaces.
- To make the students understand life cycle of the applets and its functionality.
- To make the students understand the usage util package.
- To teach the student, to develop java programs using interfaces.

Recommended System/Software Requirements:

- Intel based desktop PC with minimum of 2.6GHZ or faster processor with at least 256 MB RAM and 40GB free disk space.
- Operating system: Flavor of any WINDOWS.
- Software:j2sdk1.7.
- Linux and MvSQL.
- Eclipse or Net beam.

INTRODUCTION TO OOP

Object-oriented programming (OOP) is a computer science term used to characterize a programming language that began development in the 1960's. The term 'object-oriented programming' was originally coined by Xerox PARC to designate a computer application that describes the methodology of using objects as the foundation for computation. By the 1980's, OOP rose to prominence as the programming language of choice, exemplified by the success of C++. Currently, OOPs such as Java, J2EE, C++,C=, Visual Basic.NET, Python and java Script are popular OOP programming languages that any career-oriented Software Engineer or developer should be familiar with.

OOP is widely accepted as being far more flexible than other computer programming languages. OOPs use three basic concepts as the fundamentals for the Abstraction, Polymorphism, Event Handling and Encapsulation are also significant concepts within object-oriented programming languages that are explained in online tutorial describing the functionality of each concept in detail.

The java platform is undoubtedly fast moving and comprehensive. Its many application programming interfaces (APIs) provide a wealth of functionality for all aspects of application and system-level programming. Real-world developers never use one or two APIs to solve a problem, but bring together key functionality spanning a number of APIs, Knowing which APIs you need,

which parts of which APIs you need, and how the APIs work together to create the best solution can be a daunting task.

1. GUIDELINES TO STUDENTS

1. Equipment in the lab for the use of student community. Students need to maintain a proper decorum in the computer lab. Students must use the equipment with care. Any damage is caused is punishable.

2. Students are instructed to come to lab in formal dresses only.

3. Students are supposed to occupy the systems allotted to them and are not supposed to talk or make noise in the lab.

4. Students are required to carry their observation book and lab records with completed exercises while entering the lab.

5. Lab records need to be submitted every week.

6. Students are not supposed to use pen drives in the lab.

2. LAB OBJECTIVE

- To introduce Java compiler and eclipse platform.
- To make the student learn an object oriented way of solving problems using java.
- To make the students to write programs using multithreading concepts and handle exceptions.
- To make the students to write programs that connects to a database and be able to perform various operations.
- To make the students to create the Graphical User Interface using Applets, AWT Components & Swing Components.

3. LAB OUTCOME

- Able to use Java compiler and eclipse platform to write and execute java program.
- Understand and Apply Object oriented features and Java concepts.
- Able to apply the concept of multithreading and implement exception handling.
- Able to access data from a Database with java program.
- Develop applications using Console I/O and File I/O,GUI applications

4. INTRODUCTION ABOUT LAB

There are 30 systems (HP) installed in this Lab. Their configurations are as follows:

Processor	: Pentium(R) Dual-Core CPU E5700 @ 3.00GHz
RAM	: 1 GB
Hard Disk	: 320 GB
Mouse	: Optical Mouse

5. List of experiments as per the university curriculum

S.No.	Name of the Program	Page No.
1	**Week1 :**	8-9
	Use eclipse or Netbean platform and acquaint with the various menus, create a test project, add a test class and run it see how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.	
2	**Week 2 :**	10-13
	Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divide by zero.	
3	**Week 3 :**	14-16
	a) Develop an applet that displays a simple message.	
	b) Develop an Applet that receives an integer in one text field & compute its factorial value & returns it in another text filed when the button "Compute" is clicked	
4.	**Week 4 :**	17-19
	Write a program that creates a user interface to perform integer divisions. 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 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box	
5	**Week 5 :**	20-22
	Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.	
6	**Week 6 :**	23-29
	Write a java program that connects to a database using JDBC and does	

	add, deletes, modify and retrieve operations	
7	**Week 7 :**	30-34
	Write a java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with “stop” or “ready” or “go” should appear above the buttons in a selected color. Initially there is no message shown.	
8	**Week 8 :**	35-36
	Write a java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.	
9	**Week 9 :**	37-39
	Suppose that a table named Table.txt is stored in a text file. The first line in the file header and the remaining lines correspond to row in the table. The elements are separated by commas. Write a Java program to display the table using labels in grid layout.	
10	**Week 10 :**	40-42
	Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired. (Use adapter classes).	
11	**Week 11 :**	43-45
	Write a java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t).it takes a name or phone number as input and prints the corresponding other value from the hash table(hint: use hash tables)	
12	**Week 12 :**	46-50
	Implement the above program with database instead of a text file.	
13	**Week 13 :**	51-54
	Write a java program that takes tab separated data (one record per line) from a text file and inserts them into a database	
14	**Week 14 :**	55-57
	Write a java program that prints the meta-data of a given table.	

6. List of Additional Experiments

1	Write a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions?	58-59
2	The Fibonacci sequence is defined by the following rule. The first 2 values in the sequence are 1, 1. Every subsequent value is the sum of the 2 values preceding it. Write a Java program that uses both recursive and non-recursive functions to print the nth value of the Fibonacci sequence?	60-62
3	Write a Java program that prompts the user for an integer and then prints out all the prime numbers up to that Integer?	63
4	Write a Java program that checks whether a given string is a palindrome or not. Ex: MADAM is a palindrome?	64
5	Write a Java program for sorting a given list of names in ascending order?	65-66
6	Write a Java program to multiply two given matrices?	67-68
7	Write a Java program that reads a line of integers and then displays each integer and the sum of all integers. (use StringTokenizer class)?	69-70
8	Write a Java program that reads on file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes?	71
9	Write a Java program that reads a file and displays the file on the screen, with a line number before each line?	72
10	Write a Java program that displays the number of characters, lines and words in a text?	73-74

Solutions:-

1. Use eclipse or Netbean platform and acquaint with the various menus, create a test project, add a test class and run it see how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.

Program:-

```
public class Prog1
 {

        public static void main(String[] args)
        {
                System.out.println("\n Prog. is showing even no");
                for(int i=2;i<=20;i++)
                {
                        if(i%2==0)
                        {
                                System.out.print("\n "+i);
                        }

                }
        }
}
```

Compile:-

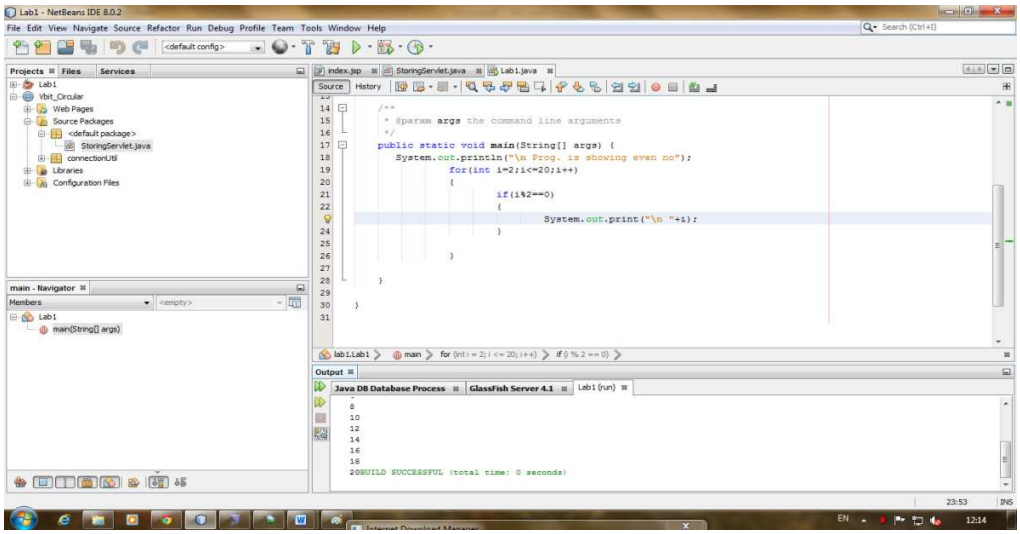
D:>javac Prog1.java

Run:-

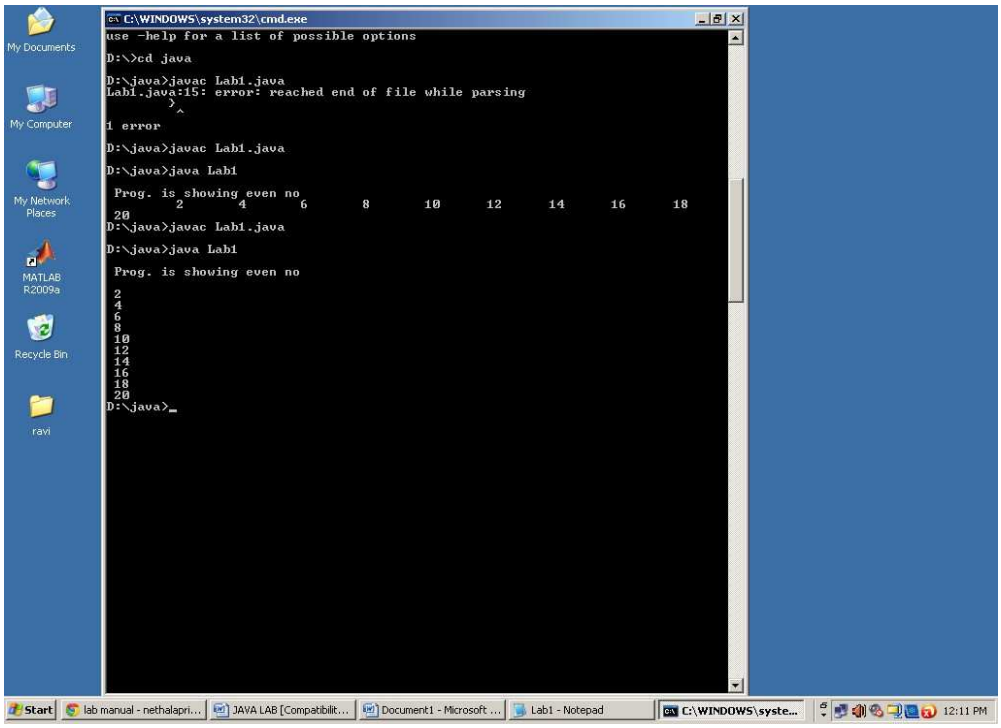
D:>java Prog1

Output:-

In Netbeans IDE:-



In Command Prompt:-



2. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divide by zero.

Program:-

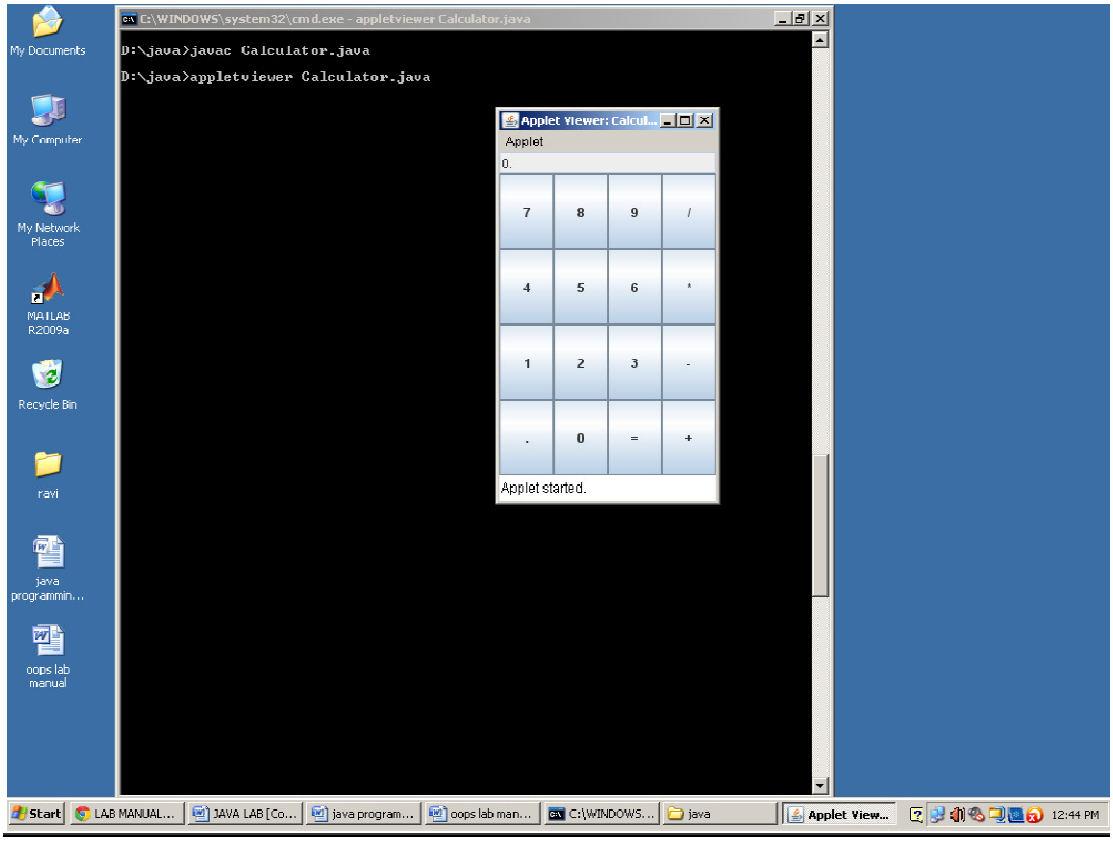
```
import javax.swing.*;
import java.awt.*;
import  java.awt.event.*;
//<applet code=Calculator height=300 width=200></applet>
public class Calculator extends JApplet
 {
   public void init()
    {
      CalculatorPanel calc=new CalculatorPanel();
      getContentPane().add(calc);
       }
    }
   class CalculatorPanel extends JPanel implements ActionListener
 {
      JButton n1,n2,n3,n4,n5,n6,n7,n8,n9,n0,plus,minus,mul,div,dot,equal;
      static JTextField result=new JTextField("0",45);
      static String lastCommand=null;
      JOptionPane p=new JOptionPane();
      double preRes=0,secVal=0,res;
      private static void assign(String  no)
        {
         if((result.getText()).equals("0"))
            result.setText(no);
          else if(lastCommand=="=")
           {
            result.setText(no);
            lastCommand=null;

           }

          else

            result.setText(result.getText()+no);

         }

      public CalculatorPanel()
       {
         setLayout(new BorderLayout());
         result.setEditable(false);

         result.setSize(300,200);

         add(result,BorderLayout.NORTH);

         JPanel panel=new JPanel();

         panel.setLayout(new  GridLayout(4,4));
```

```
    n7=new JButton("7");
    panel.add(n7);
    n7.addActionListener(this);
    n8=new JButton("8");
    panel.add(n8);
    n8.addActionListener(this);
    n9=new JButton("9");
    panel.add(n9);
    n9.addActionListener(this);
    div=new JButton("/");
    panel.add(div);
    div.addActionListener(this);
    n4=new JButton("4");
    panel.add(n4);
    n4.addActionListener(this);
    n5=new JButton("5");
    panel.add(n5);
    n5.addActionListener(this);
    n6=new JButton("6");
    panel.add(n6);
    n6.addActionListener(this);
    mul=new JButton("*");
    panel.add(mul);
    mul.addActionListener(this);
    n1=new JButton("1");
    panel.add(n1);
    n1.addActionListener(this);
    n2=new JButton("2");
    panel.add(n2);
    n2.addActionListener(this);

    n3=new  JButton("3");
    panel.add(n3);
    n3.addActionListener(this);
    minus=new JButton("-");
    panel.add(minus);
    minus.addActionListener(this);
    dot=new JButton(".");
    panel.add(dot);
    dot.addActionListener(this);
    n0=new  JButton("0");
    panel.add(n0);
    n0.addActionListener(this);
    equal=new JButton("=");
    panel.add(equal);
    equal.addActionListener(this);
    plus=new JButton("+");
    panel.add(plus);
    plus.addActionListener(this);
    add(panel,BorderLayout.CENTER);
  }
  public void actionPerformed(ActionEvent ae)
```

```
   {
if(ae.getSource()==n1) assign("1");
else  if(ae.getSource()==n2)  assign("2");
else  if(ae.getSource()==n3)  assign("3");
else  if(ae.getSource()==n4)  assign("4");
else  if(ae.getSource()==n5)  assign("5");
else  if(ae.getSource()==n6)  assign("6");
else  if(ae.getSource()==n7)  assign("7");
else  if(ae.getSource()==n8)  assign("8");
else  if(ae.getSource()==n9)  assign("9");
else  if(ae.getSource()==n0)  assign("0");
else if(ae.getSource()==dot)
      {
      if(((result.getText()).indexOf("."))==-1)
         result.setText(result.getText()+".");
     }
else if(ae.getSource()==minus)
      {
      preRes=Double.parseDouble(result.getText());
      lastCommand="-";
      result.setText("0");
      }
else if(ae.getSource()==div)
      {
      preRes=Double.parseDouble(result.getText());
      lastCommand="/";
      result.setText("0");
      }
else if(ae.getSource()==equal)
      {
      secVal=Double.parseDouble(result.getText());
      if(lastCommand.equals("/"))
          res=preRes/secVal;
      else if(lastCommand.equals("*"))
          res=preRes*secVal;
      else if(lastCommand.equals("-"))
          res=preRes-secVal;
      else if(lastCommand.equals("+"))
          res=preRes+secVal;
      result.setText(" "+res);
      lastCommand="=";
      }
      else if(ae.getSource()==mul)
      {
       preRes=Double.parseDouble(result.getText());
       lastCommand="*";
       result.setText("0");
       }
else if(ae.getSource()==plus)
       {
       preRes=Double.parseDouble(result.getText());
       lastCommand="+";
```

```
            result.setText("0");
            }
        }
    }
```

Output:-

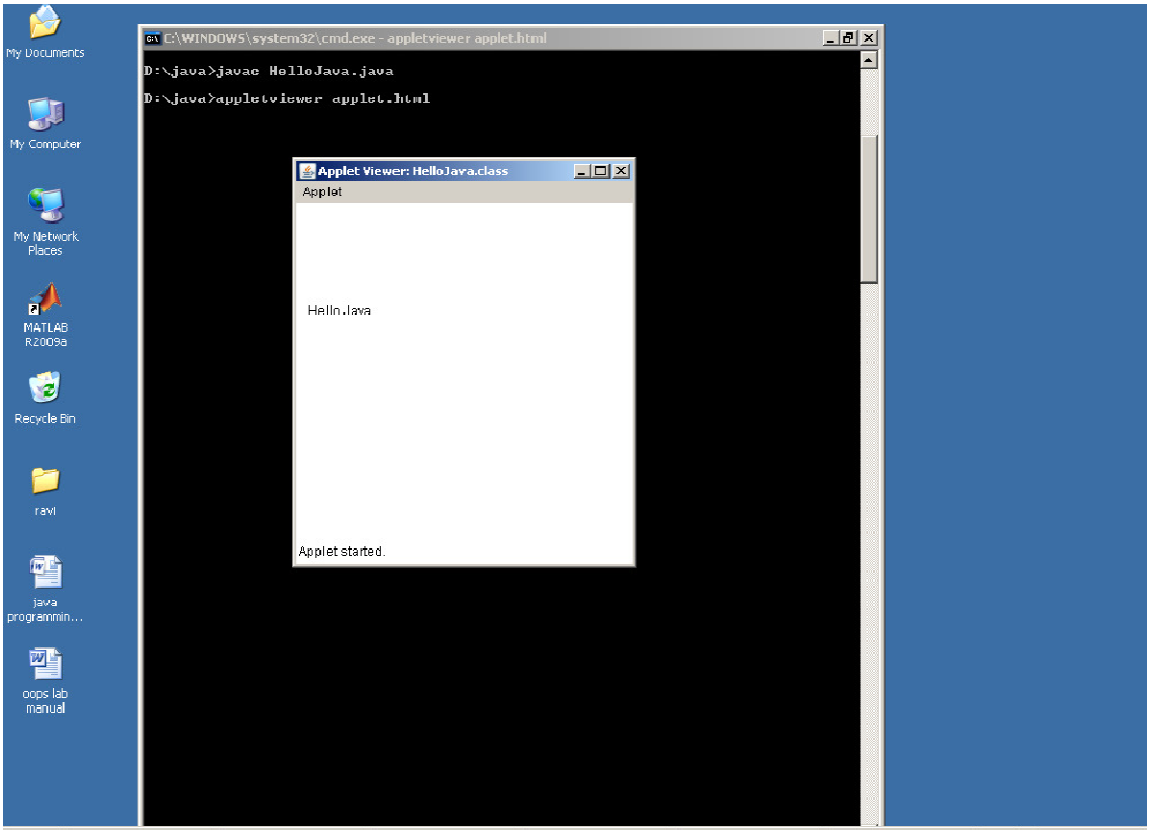


3. a) Develop an applet that displays a simple message.

Program:-

```
import java.awt.*;
import java.applet.*;
/*<applet  code = "HelloJava"          width = 200     height = 60 > </applet>*/
public class HelloJava extends Applet {
public void paint(Graphics g) {
                    g.drawString("Hello Java", 10, 100);
          }
}
```

Output:-



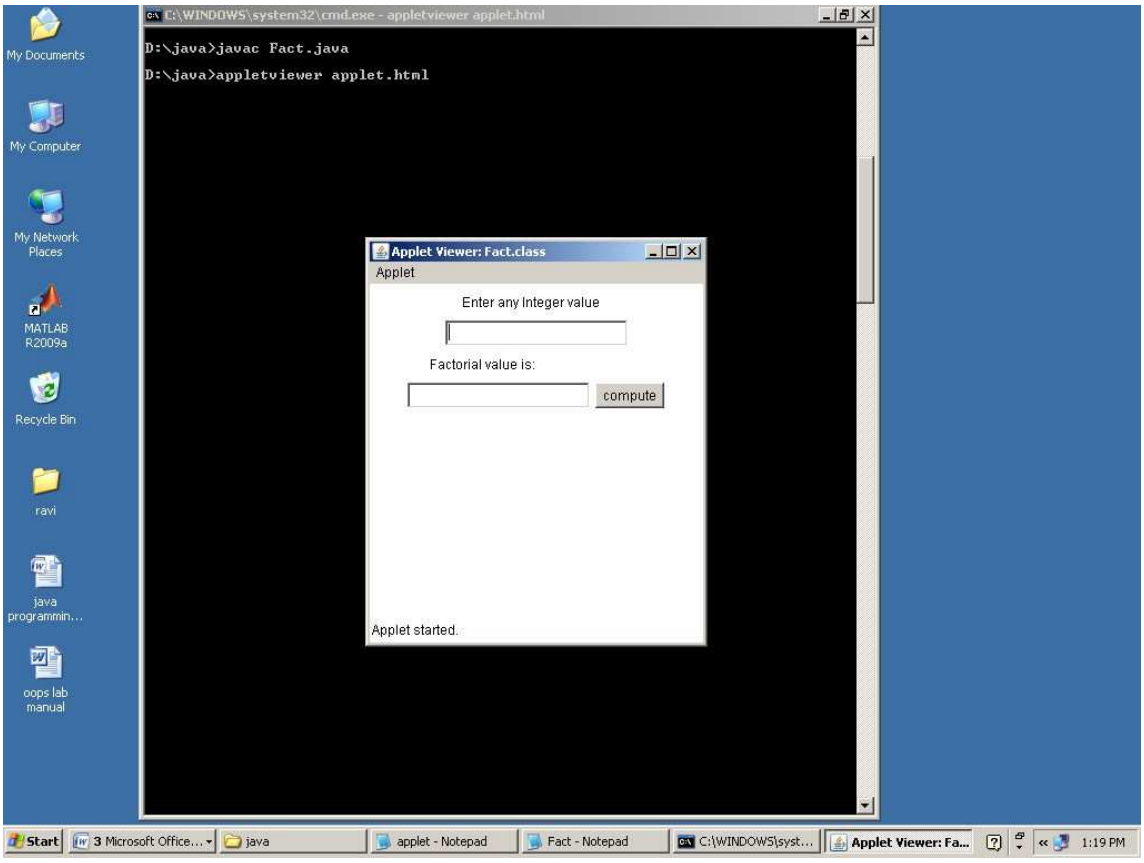
3.**b) Develop an Applet that receives an integer in one text field & compute its factorial value & returns it in another text filed when the button "Compute" is clicked.**

Program:-

```
import java.awt.*;

import java.lang.String;

import  java.awt.event.*;

import java.applet.Applet;

public class Fact extends Applet implements ActionListener

{

String str;

Button b0;

        TextField t1,t2;

        Label l1;

         public void init(){

         Panel p=new Panel();

         p.setLayout(new GridLayout());

         add(new Label("Enter any Integer value"));

         add(t1=new TextField(20));

         add(new Label("Factorial  value is:        "));

         add(t2=new TextField(20));

         add(b0=new Button("compute"));

         b0.addActionListener(this);

       }

        public void actionPerformed(ActionEvent e)

         {

                int i,n,f=1;


                 n=Integer.parseInt(t1.getText());
```

```
            for(i=1;i<=n;i++)
            f=f*i;
            t2.setText(String.valueOf(f));
          repaint();
    }
}
```

Output:-



4. Write a program that creates a user interface to perform integer divisions. 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 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

Program:-

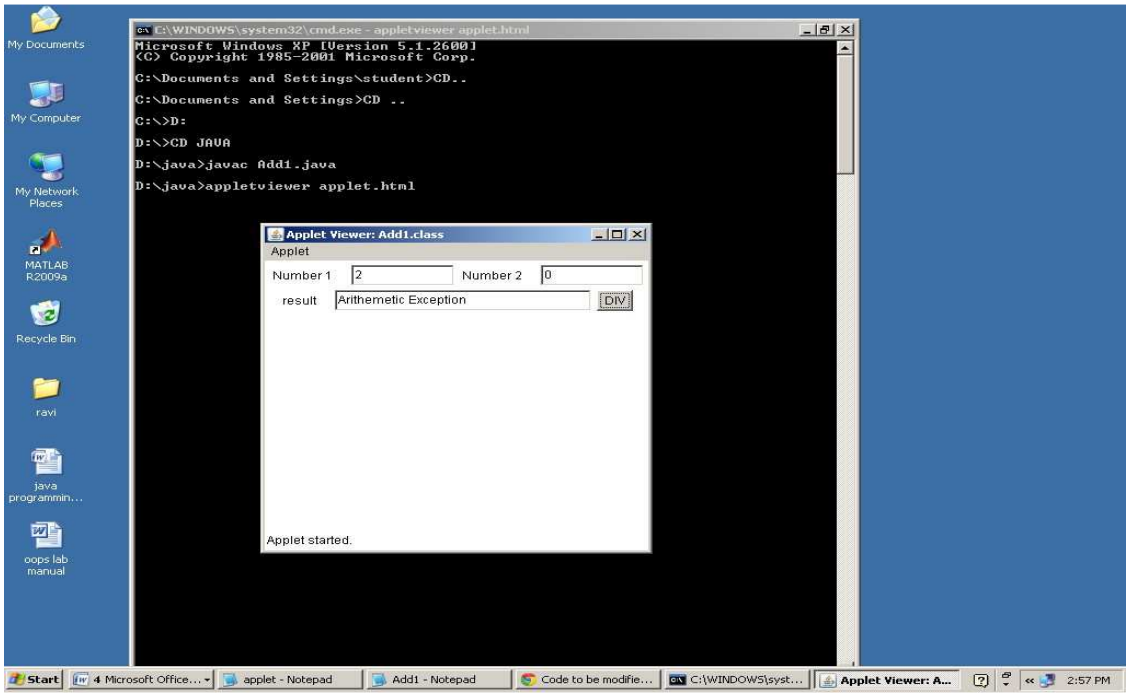
```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class Add1 extends Applet implements ActionListener
{
   String msg;
   TextField num1, num2, res;
   Label l1, l2, l3;
   Button div;
    public void  init()
 {
       l1 = new Label("Number 1");
      l2 = new Label("Number 2");
      l3 = new Label("result");
      num1 = new TextField(10);
      num2 = new TextField(10);
      res = new  TextField(30);
     div = new Button("DIV");
     div.addActionListener(this);
     add(l1);
     add(num1);
    add(l2);
    add(num2);
    add(l3);
    add(res);
    add(div);
}

public void actionPerformed(ActionEvent ae)
{
       String arg = ae.getActionCommand();
       if (arg.equals("DIV"))
 {
           String s1 = num1.getText();
           String s2 = num2.getText();
           int num1 = Integer.parseInt(s1);
```

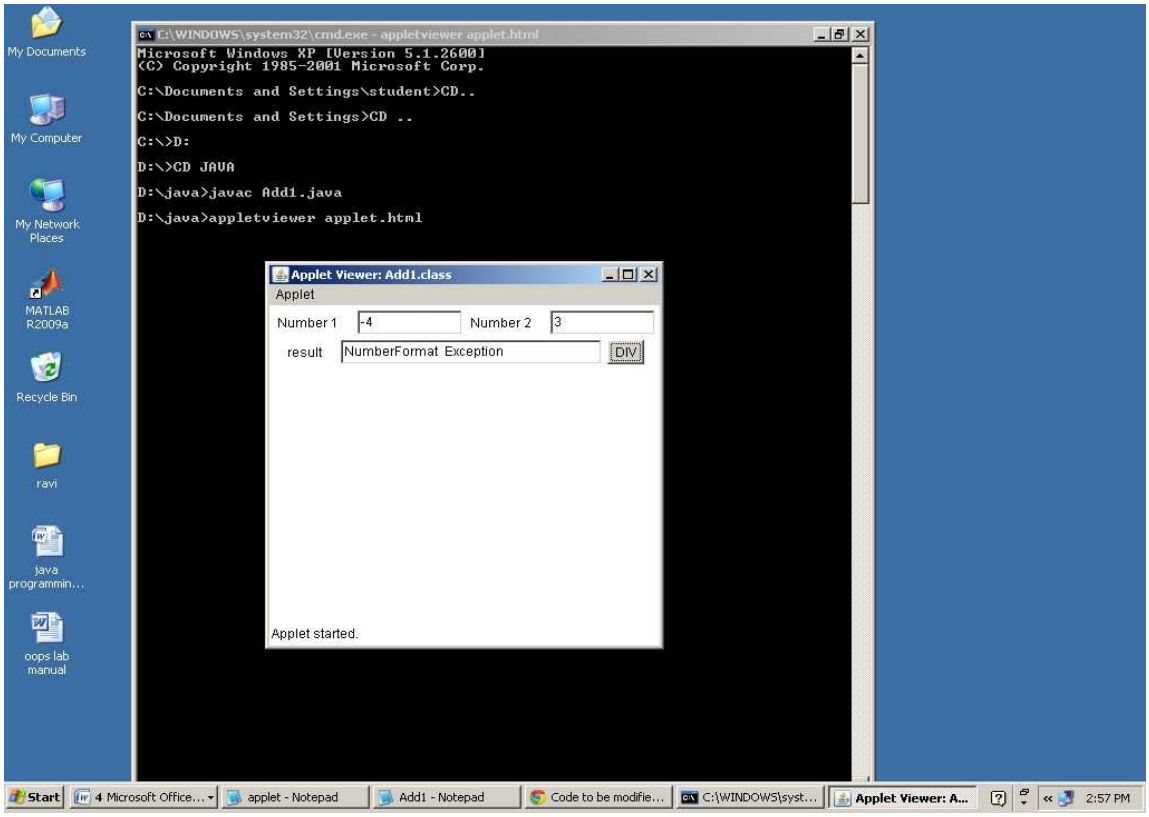
```
            int num2 = Integer.parseInt(s2);
            if (num2 == 0)
             {
            msg = "Arithemetic Exception ";
            repaint();
            }
             else if ((num1 < 0) || (num2 < 0))
            {
            msg = "NumberFormat Exception";
            repaint();
            }
            else
            {
            int num3 = num1 / num2;
            msg = String.valueOf(num3);
            }
            res.setText(msg);
        }
    }
public void paint(Graphics g)
 {
        //g.drawString(msg, 30, 70);
    }
}
```

APPLET.HTML

```
<html>

<head>

</head>

<body>

/*<applet code="Add1.class"width=350 height=300>

</applet>*/

</body>

</html>
```

Output:-


C:\WINDOWS\system32\cmd.exe - appletviewer applet.html
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\student>CD..
C:\Documents and Settings>CD ..
C:\>D:
D:\>CD JAVA
D:\java>javac Add1.java
D:\java>appletviewer applet.html
Applet Viewer: Add1.class
Applet
Number 1
2
Number 2
0
result
Arithemetic Exception
DIV
Applet started.


C:\WINDOWS\system32\cmd.exe - appletviewer applet.html
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\student>CD..
C:\Documents and Settings>CD ..
C:\>D:
D:\>CD JAVA
D:\java>javac Add1.java
D:\java>appletviewer applet.html
Applet Viewer: Add1.class
Applet
Number 1
-4
Number 2
3
result
NumberFormat Exception
DIV
Applet started.

5.) Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

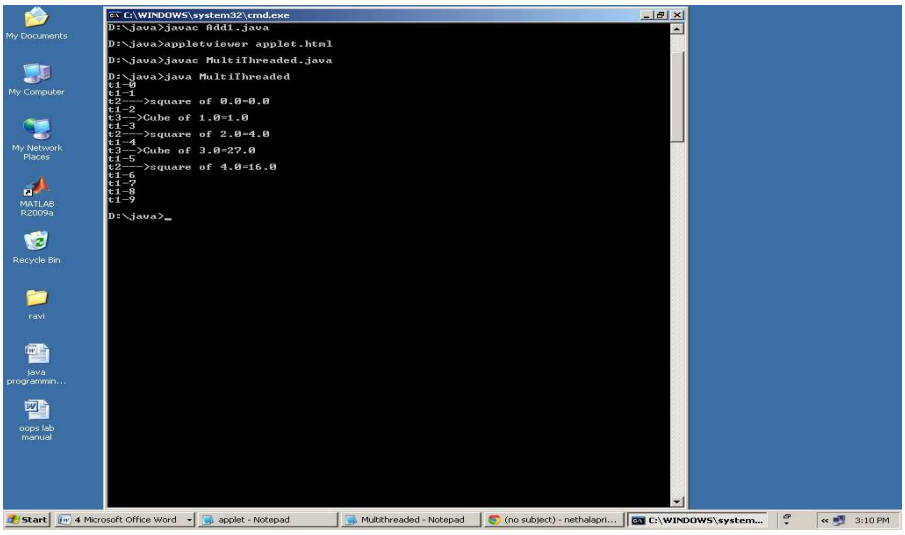
Program:-

```
class RandomGenThread implements Runnable
{
    double num;
    public void run()
     {
            try {
                    SquareThread sqt = new SquareThread();
                    Thread squareThread = new Thread(sqt);
                    CubeThread cbt = new CubeThread();
                    Threadcube Thread = new Thread(cbt);
                    squareThread.start();
                    cubeThread.start();
                    for(int  i=0;i<10;i++)
                    {
                            System.out.println("t1-"+i);
                            if(i%2 == 0)
                             {
                            sqt.setNum(new Double(i));
                             }
                            else
                            {
                                    cbt.setNum(new Double(i));
                            }
                                    Thread.sleep(1000);
            }
            } catch (InterruptedException e)
                    {
                    e.printStackTrace();
                    }
    }

}
 class SquareThread implements Runnable
 {
    Double num;
    public void run()
 {

            try {
```

```
                int i=0;
                do{
                        i++;
                        if(num != null&&num %2 ==0)
                         {
                        System.out.println("t2--->square of "+num+"="+(num*num));
                        num = null;
                        }
                Thread.sleep(1000);
                }while(i<=5);
        }
                 catch (Exception e)
                 {
                        e.printStackTrace();
        }
        }

    public Double getNum()
    {
        return num;
    }

    public void setNum(Double num)
    {
        this.num = num;
    }
}

class CubeThread implements Runnable
 {
    Double num;
    public void run()
    {
        try {
                int i=0;
                do{
                        i++;
                        if(num != null&&num%2 !=0)
                        {
                        System.out.println("t3-->Cube of "+num+"="+(num*num*num));
                        num=null;
                }
                Thread.sleep(1000);
                }
                while(i<=5);
```

```
            }
                        catch (Exception e)
                        {
                           e.printStackTrace();
            }
            }
    public Double getNum()
    {
            return num;
    }

    public void setNum(Double num)
    {
            this.num =  num;
    }
}
public  class MultiThreaded
 {
    public static void main(String[] args) throws InterruptedException
{
            Thread randomThread = new Thread(new RandomGenThread());
            randomThread.start();
    }

}
```

Output:-



6).Write a java program that connects to a database using JDBC and does add, deletes, modify and retrieve operations?

Program:-

ConnectionUtil.java

```
import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class ConnectionUtil

 {


        public static Connection getConnection() throws SQLException

         {

                Connection connection = null;

                try

                 {

Class.forName("com.mysql.jdbc.Connection");

connection = DriverManager.getConnection("jdbc:mysql://192.168.216.250:3306/vijju","root", "root");

                }

           catch (ClassNotFoundException e)

              {

                        e.printStackTrace();

                }

                         catch (SQLException e)

                        {

                        e.printStackTrace();

                        }

                System.out.println("message for connection open"+connection);

                return connection;
```

```
        }
         }
```

StudentDetails.java

```
public class StudentDetails
{
        private long st_id;
        private String st_name;
        private long st_mobile;
        public StudentDetails(long st_id,String st_name,long st_mobile)
 {
                this.st_id = st_id;
                this.st_name = st_name;
                this.st_mobile = st_mobile;
}
        public long getSt_id()
         {
                return st_id;
        }
        public void setSt_id(long st_id)
         {
                this.st_id = st_id;
        }
        public String getSt_name()
         {
                return st_name;
        }
        public void setSt_name(String st_name)
        {
```

```
		this.st_name = st_name;
	}
	public long getSt_mobile()
	{
		return st_mobile;
	}
	public void setSt_mobile(long st_mobile)
	{
		this.st_mobile = st_mobile;
	}
}
```

Prog6.java

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
public class Prog6
{
String createTableQuery = "create table test.student_details (st_namevarchar(50), st_mobile numeric(10), st_id numeric(10))";
	public static void main(String[] args)
{		Connection conn = null;
		try
 {
			conn = ConnectionUtil.getConnection();
			Prog6 prog6 = new Prog6();
			System.out.println("Student_details table data before inserting:");
			prog6.retrieveData(conn);
```

```
        StudentDetails st1 = new StudentDetails(1, "GNIT", 23232323);
        StudentDetails st2 = new StudentDetails(2, "GNEC", 24242424);
        StudentDetails st3 = new StudentDetails(3, "GNITC", 25252525);
        prog6.insertData(conn, st1);
        prog6.insertData(conn, st2);
        prog6.insertData(conn,  st3);
        System.out.println("Student_details table data after inserting:");
        prog6.retrieveData(conn);
        prog6.deleteARow(conn,2);
        System.out.println("Student_details table data after deleting:");
        prog6.retrieveData(conn);
        prog6.modifyData(conn, 26262626, 3);
        System.out.println("Student_details table data after modifying:");
        prog6.retrieveData(conn);
} catch (SQLException e)
         {
        e.printStackTrace();
        } finally
         {
        try {
        conn.close();
        } catch (SQLException e)
         {
                e.printStackTrace();
        }
        }
        }
private void modifyData(Connection conn, long st_mobile, long st_id)
```

```
        {
        String query = "update student_details set st_mobile=? wherest_id=?";
        try {
                PreparedStatement pstmt = conn.prepareStatement(query);
                pstmt.setLong(1, st_mobile);
                pstmt.setLong(2, st_id);
                int row = pstmt.executeUpdate();
                //System.out.println(row);
                } catch (SQLException  e)
                {                       e.printStackTrace();
                }
                }
  private void deleteARow(Connection conn, long st_id)
         {
        String query = "delete from student_details where st_id=?" ;
        try
                {
                PreparedStatement pstmt = conn.prepareStatement(query);
                pstmt.setLong(1, st_id);
                int row = pstmt.executeUpdate();
                //System.out.println(row);
                 }
                catch (SQLException e)
                 {
                        e.printStackTrace();
        }
}
private void insertData(Connection conn, StudentDetails details)
```

```
{
        String query = "insert into test.student_details values (?,?,?)";
        try
                {
                PreparedStatement pstmt = conn.prepareStatement(query);
                pstmt.setString(1,details.getSt_name());
                pstmt.setLong(2, details.getSt_mobile());
                pstmt.setLong(3, details.getSt_id());
                int row = pstmt.executeUpdate();
                //System.out.println(row);
                }
                catch (SQLException e)
                {
                e.printStackTrace();
        }
        }
    private void retrieveData(Connection conn)
         {
        try {
                String query = "select * from test.student_details";
                PreparedStatement pstmt = conn.prepareStatement(query);
                ResultSet rs = pstmt.executeQuery();
                System.out.println("ST_ID\tST_NAME\t\tST_MOBILE");
                while(rs.next())
                {
    System.out.println(rs.getString("st_id")+"\t"+rs.getString("st_name")+"\t\t"+rs.getString(
"st_mobile"));
                }
```

```
            }
            catch (SQLException e)
             {
                    e.printStackTrace();
            }
            }
        }
```

Output:-

```
C:\Windows\system32\cmd.exe
Student_details table data before inserting:
ST_ID   ST_NAME         ST_MOBILE
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
3       GNITC           25262625
Student_details table data after inserting:
ST_ID   ST_NAME         ST_MOBILE
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
2       GNEC            24242424
3       GNITC           25252525
Student_details table data after deleting:
ST_ID   ST_NAME         ST_MOBILE
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
3       GNITC           25262625
1       GNIT            23232323
3       GNITC           25262625
```

7) Write a java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with "stop" or "ready" or "go" should appear above the buttons in a selected color. Initially there is no message shown.

Program:-

TrafficSignal.java

```
import java.awt.*;

import java.applet.*;

import  java.awt.event.*;

public class TrafficSignal extends Applet implements Runnable

 {

    Thread t;

    Font f, f1;

    int i = 0, a = 0, j = 0;

    public void init()

 {

        setBackground(Color.lightGray);

        f = new Font("TimesNewRoman", f.ITALIC, 28);

        f1 = new Font("TimesNewRoman", Font.ITALIC + Font.BOLD, 28);

    }

    public void start()

{

        t = new Thread(this);

        t.start();

  }

    public void run()

 {

        for (i = 10; i >= 0; i--)//countdown

        {
```

```
    try
{
        Thread.sleep(1000);
    }
    catch (Exception e)
     {
         System.out.println(e);
    }
     if (i <= 10 && i > 3)//red
    {
        a = 1;
        repaint();
    }
    else if (i <= 3 && i > 0)//yellow
    {
        a = 2;
        repaint();
     }
     else if (i == 0)//green
    {
         for (j = 0; j < 10; j++)
                {
                        a = 3;
                 try
                {
                         Thread.sleep(1000);
                 }
                        catch (Exception e)
```

```
                    {
                    System.out.println(e);
                    }
            repaint();
        }
                if (j == 10)//end of green(return to red)
                {
                    run();
                }
                }
                }
    repaint();
  }
  public void paint(Graphics g)
{
    setBackground(Color.lightGray);//ROAD
    g.setColor(Color.black);//POLE UP
    g.fillRect(150, 150, 50, 150);
    g.drawRect(150, 150, 50, 150);
    g.setColor(Color.black);//POLE DOWN
    g.fillRect(165, 300, 20, 155);
    g.drawRect(165, 300, 20, 155);
    g.drawOval(150, 150, 50, 50);//RED
    g.drawOval(150, 200, 50, 50);//YELLOW
    g.drawOval(150, 250, 50, 50);//GREEN
    g.setColor(Color.red);//COUNTDOWN STOP
    g.setFont(f);
    g.drawString("" + i, 50, 50);
```

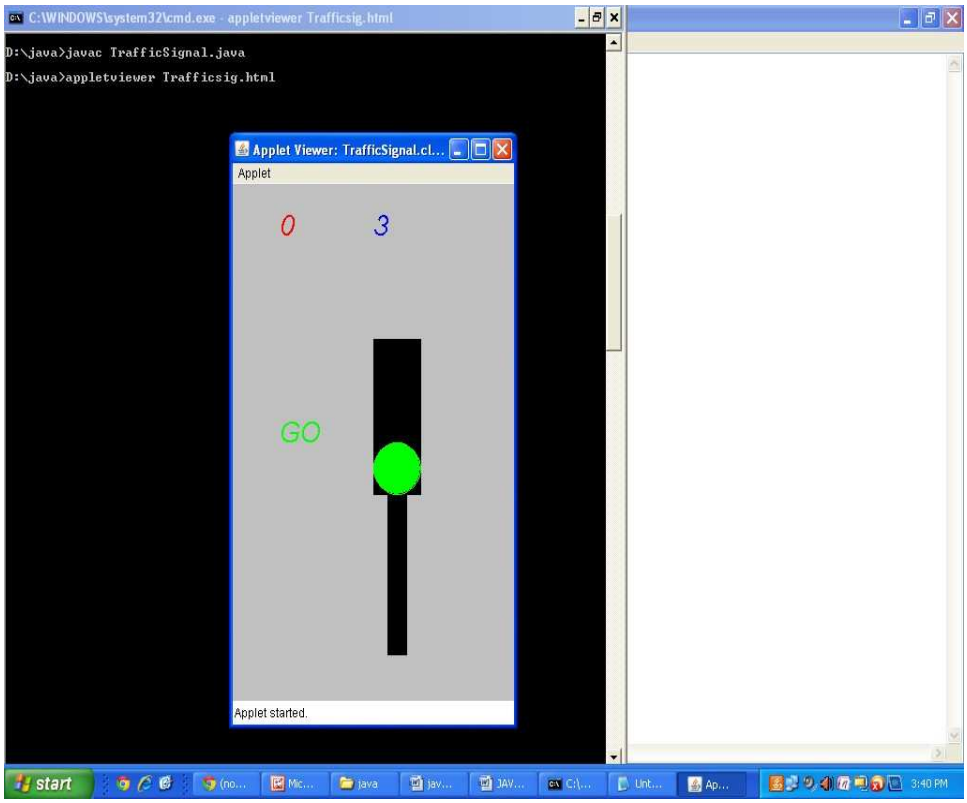
```
 if (a == 1)//REDSIGNAL

 {
    g.setColor(Color.red);
    g.fillOval(150, 150, 50, 50);

    g.drawOval(150, 150, 50, 50);

    g.drawString("STOP", 50, 150);

}

 if (a == 2)//YELLOWSIGNAL

 {

    g.setColor(Color.yellow);

    g.fillOval(150, 200, 50, 50);

    g.drawOval(150, 200, 50, 50);

    g.drawString("READY", 50, 200);

 }

 if (a == 3)//GREENSIGNAL

 {
    g.setColor(Color.blue);//countdown
    g.setFont(f);

    g.drawString("" + j, 150, 50);

    g.setColor(Color.green);

    g.fillOval(150, 250, 50, 50);

    g.drawOval(150, 250, 50, 50);

    g.drawString("GO",  50, 250);
 }
 int x1[] = {220, 300, 300,  280};
 int y1[] = {250, 150, 250,  150};

 int n1 = 4;

 int n2 = 3;

 int x2[] = {340, 380,  380};

 int y2[] = {150, 100,  150};

 int x3[] = {460, 460,  500};
```

```
        int y3[] = {150, 100, 150};

    }

}
```

TrafficSignal.html

```
<html>
<head>
</head>
<body>
/*<applet code="TrafficSignal.class" height=500 width=300></applet>*/
</body>
</html>
```

Output:-



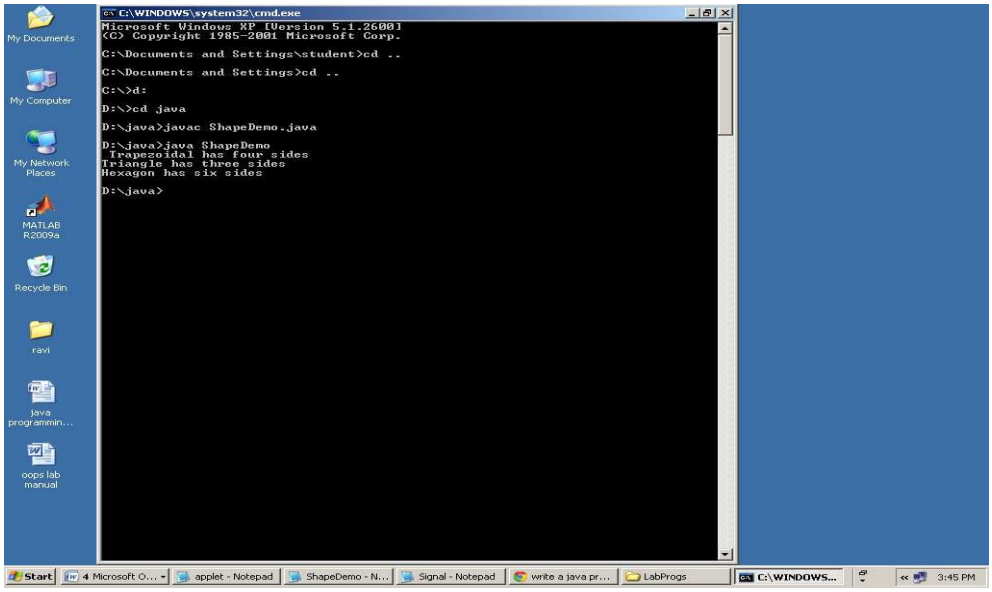
8) Write a java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.

Program:-

```
abstract class Shape

{

 abstract void numberOfSides();

  }

class Trapezoid extends Shape

{

void numberOfSides()

 {

          System.out.println(" Trapezoidal has four sides");

}

  }

class Triangle extends Shape

{

 void numberOfSides()

{

 System.out.println("Triangle has three sides");

}

  }

 class Hexagon extends Shape

{

void numberOfSides()

{
System.out.println("Hexagon has six sides");
 }
 }
class ShapeDemo
```

```
{
 public static void main(String args[ ])
 {
Trapezoid t=new Trapezoid();
Triangle r=new Triangle();
Hexagon h=new Hexagon();
Shape s;
s=t;
s.numberOfSides();
s=r;
s.numberOfSides();
s=h;
s.numberOfSides();
}
}
```

Output:-



9) Suppose that a table named Table.txt is stored in a text file. The first line in the file header and the remaining lines correspond to row in the table. The elements are separated by commas. Write a Java program to display the table using labels in grid layout.

Program:-

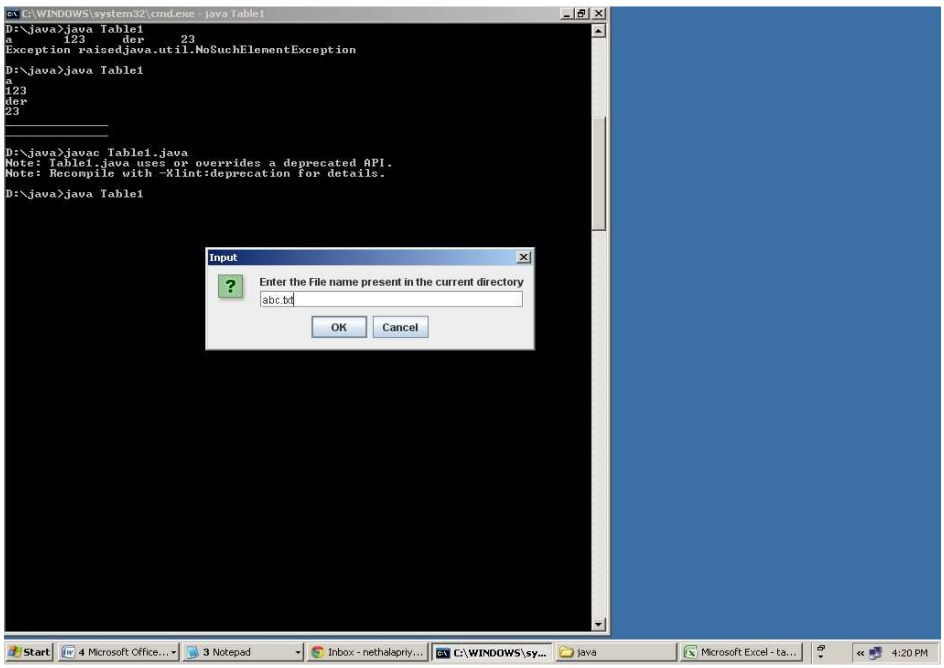
```
import java.awt.*;

import java.awt.event.*;

import javax.swing.*;

import java.util.*;

import java.io.*;

public class Table1 extends JFrame

{

int i=0;

int j=0,k=0;

Object data[][]=new Object[5][4];

Object list[][]=new Object[5][4];

JButton save;

JTable table1;

FileInputStream fis;

DataInputStream dis;

public Table1()

{

String d= " ";

Container con=getContentPane();

con.setLayout(new BorderLayout());

final String[] colHeads={"Name","Roll Number","Department","Percentage"};

try

{

String s=JOptionPane.showInputDialog("Enter the File name present in the current directory");
FileInputStream fis=new FileInputStream(s);
DataInputStream dis = new DataInputStream(fis);
```

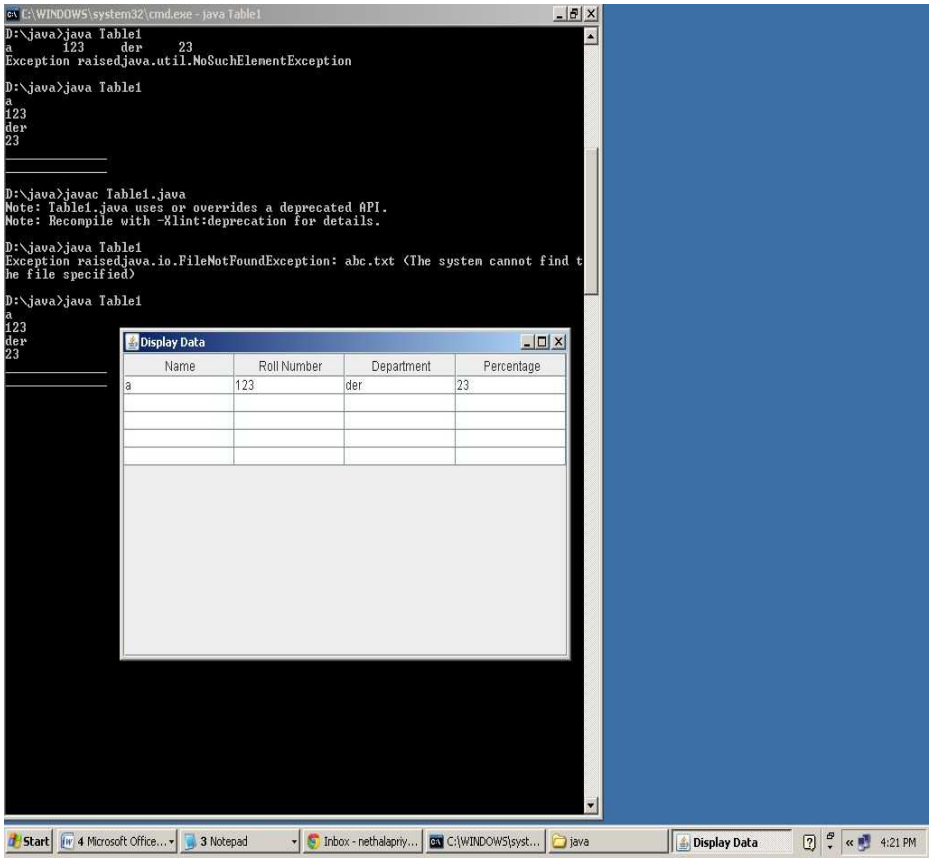
```
while ((d=dis.readLine())!=null)
{
StringTokenizer st1=new StringTokenizer(d,",");
while (st1.hasMoreTokens())
{
for (j=0;j<4;j++)
{
data[i][j]=st1.nextToken();
System.out.println(data[i][j]);
}
i++;
}
System.out.println ("_________________");
}
} catch (Exception e)
{
System.out.println ("Exception raised" +e.toString());
}
table1=new JTable(data,colHeads);
int v=ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED;
int h=ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED;
JScrollPane scroll=new JScrollPane(table1,v,h);
con.add(scroll,BorderLayout.CENTER);
}
public static void main(String args[])
{

Table1 t=new Table1();

t.setBackground(Color.green);

t.setTitle("Display Data");

t.setSize(500,300);

t.setVisible(true);

t.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

} }
```

Abc.txt:-

```
a,123,der,23

b,456,frg,45
```

Output:-





10. Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired. (Use adapter classes).

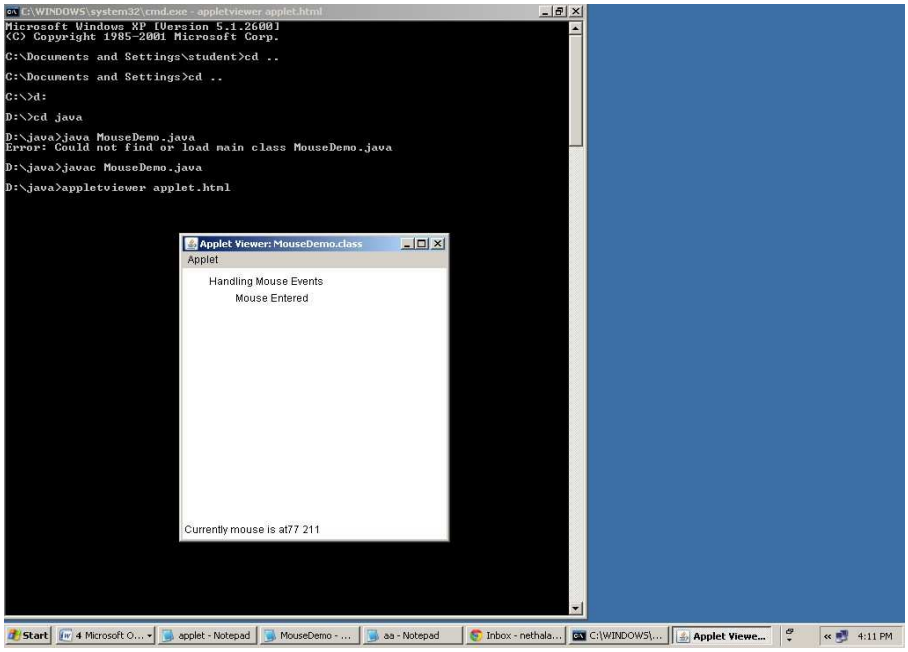
Program:-

```
import java.awt.*;

import java.applet.*;

import  java.awt.event.*;

/*<applet code="MouseDemo" width=300 height=300>

</applet>*/

public class MouseDemo extends Applet implements MouseListener,MouseMotionListener

{

int mx=0;

int my=0;

String msg="";

public void init()

{

addMouseListener(this);

addMouseMotionListener(this);

}

public void mouseClicked(MouseEvent me)

{

mx=20;

my=40;

msg="Mouse Clicked";

repaint();

}

public void mousePressed(MouseEvent me)

{

mx=30;

my=60;

msg="Mouse Pressed";
```

```
repaint();
}
public void mouseReleased(MouseEvent me)
{
mx=30;
my=60;
msg="Mouse Released";
repaint();
}
public void mouseEntered(MouseEvent me)
{
mx=40;
my=80;
msg="Mouse Entered";
repaint();
}
public void mouseExited(MouseEvent me)
{
mx=40;
my=80;
msg="Mouse Exited";
repaint();
}
public void mouseDragged(MouseEvent me)
{
mx=me.getX();
my=me.getY();
showStatus("Currently mouse dragged"+mx+" "+my);
```

```
repaint(); }

public void mouseMoved(MouseEvent me)
{
mx=me.getX();
my=me.getY();
showStatus("Currently mouse is at"+mx+" "+my);
repaint();
}
public void paint(Graphics g)
{
g.drawString("Handling Mouse Events",30,20);
g.drawString(msg,60,40);
}
}
```

Output:-



11).Write a java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t).it takes a name or phone number as input and prints the corresponding other value from

the hash table(hint: use hash tables)

Program:-

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.Hashtable;
import java.util.Iterator;
import java.util.Set;
public class HashTab
 {
        public static void main(String[] args)
        {
                HashTab prog11 = new HashTab();
                Hashtable<String, String>hashData = prog11.readFromFile("HashTab.txt");
                System.out.println("File data into Hashtable:\n"+hashData);
                prog11.printTheData(hashData, "vbit");
                prog11.printTheData(hashData, "123");
                prog11.printTheData(hashData, "---- ");
        }
        private void printTheData(Hashtable<String, String>hashData, String input)
 {
                String output = null;
                if(hashData != null)
                {
                        Set<String> keys = hashData.keySet();
                        if(keys.contains(input))
```

```
{
                output = hashData.get(input);
        }
else
{
                Iterator<String> iterator = keys.iterator();
                while(iterator.hasNext()) {
                        String key = iterator.next();
                        String value = hashData.get(key);
                        if(value.equals(input))
                        {
                                output = key;
                                break;
                        }       }       }   }
System.out.println("Input given:"+input);
if(output != null)
 {
        System.out.println("Data found in HashTable:"+output);
        }
else {
        System.out.println("Data not found in HashTable");
}       }

private Hashtable<String, String>readFromFile(String fileName) {
        Hashtable<String, String> hashData = new Hashtable<String, String>();
        try {
                File f = new File("D:\\java\\"+fileName);
                BufferedReader br = new BufferedReader(new FileReader(f));
                String line = null;
                while((line = br.readLine()) != null) {
                        String[] details = line.split("\t");
                        hashData.put(details[0], details[1]);
                }
        } catch (FileNotFoundException e) {
```

```
                e.printStackTrace();
        } catch (IOException e) {
                e.printStackTrace();            }
        return hashData;        } }
```

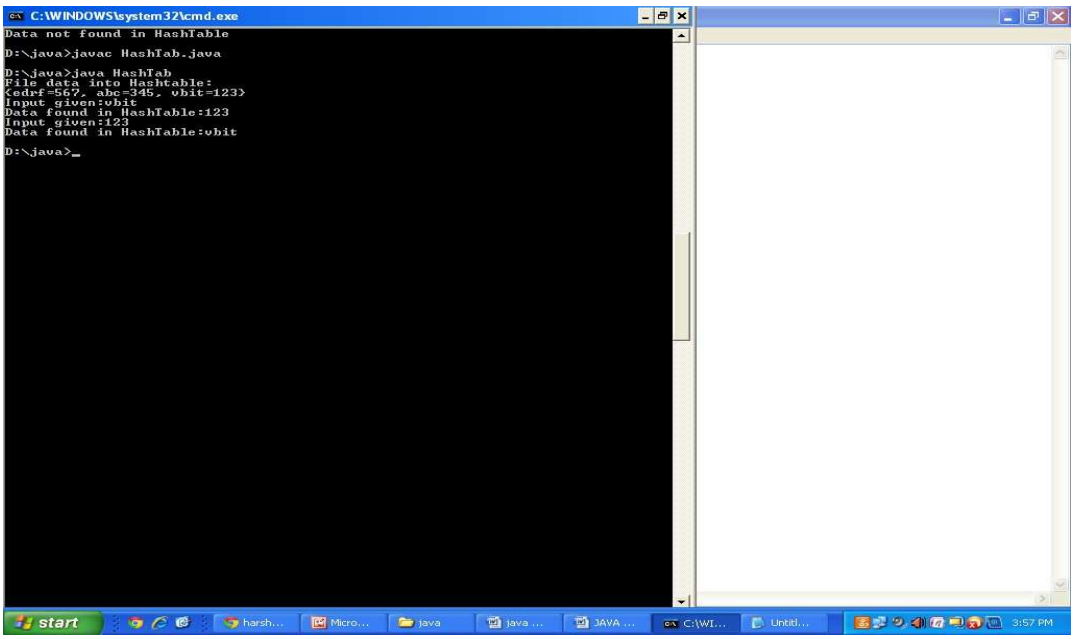
HashTab.txt

vbit 123

abc 345

edrf 567

Output:-



12. Implement the above program with database instead of a text file.

Program:-

ConnectionUtil.Java

```
package  LabProgs;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
public class ConnectionUtil
{
	public static Connection getConnection() throws SQLException
	{
		Connection connection = null;
		try
		{
			Class.forName("com.mysql.jdbc.Connection");
		connection = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/test",
						"root", "system");
		}
		catch (ClassNotFoundException e)
		{
			e.printStackTrace();
		}
		 catch (SQLException e)
		{
			e.printStackTrace();
		}
		return connection;
}
  }
```

Prog12.Java:-

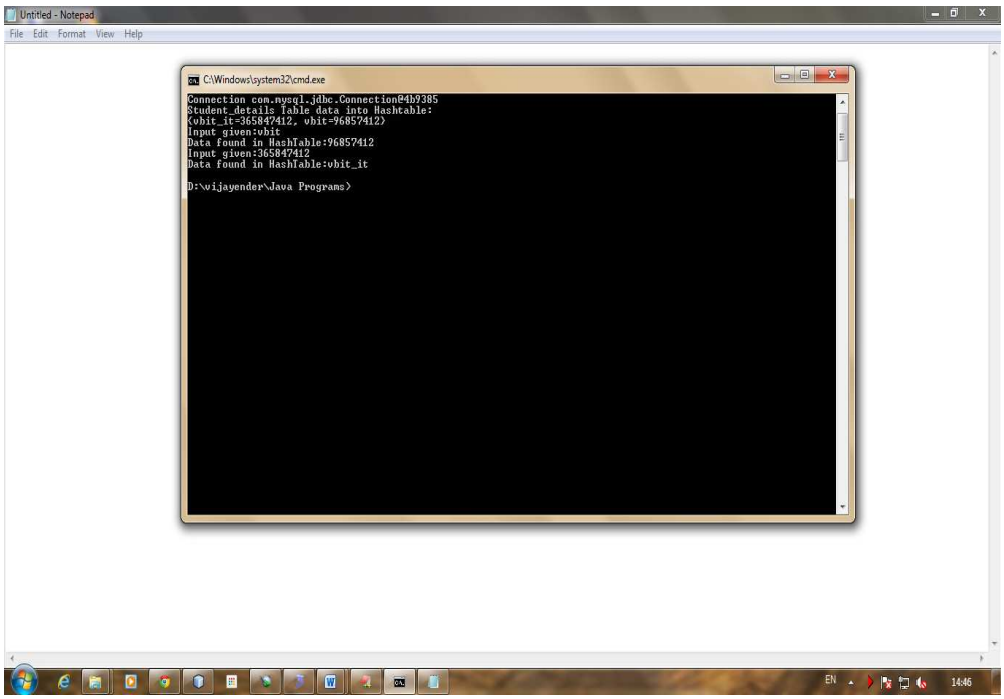
```
package LabProgs;
import java.sql.Connection;
```

```
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Hashtable;
import   java.util.Iterator;
import java.util.Set;
public class  Prog12
{
        public static void main(String[] args)
         {
                String query = "select * from test.student_details";
                Connection conn = null;
                try
                {
                        conn = ConnectionUtil.getConnection();
                        Prog12 prog12 = newProg12();
                        Hashtable<String, String>hashData = prog12.retrieveData(conn, query);
                        System.out.println("Student_details Table data into
Hashtable:\n"+hashData);
                        prog12.printTheData(hashData, "GNIT");
                        prog12.printTheData(hashData, "26262626");
                        prog12.printTheData(hashData, "****");
                }
                catch (SQLExceptione)
                {
                        e.printStackTrace();
                }
                 finally
                {
                        try
                        {
```

```
                    conn.close();
            }
            catch (SQLExceptione)
             {
                    e.printStackTrace();
            }
        }
}
Private void printTheData(Hashtable<String, String>hashData, String input)
{
        String output = null;
        if(hashData != null)
         {
            Set<String>keys = hashData.keySet();
            if(keys.contains(input)) {
                    output = hashData.get(input);
         }
        else
         {
                    Iterator<String>iterator = keys.iterator();
                    while(iterator.hasNext()) {
                            String key =  iterator.next();
                            String value = hashData.get(key);
                            if(value.equals(input)) {
                                    output = key;
                                    break;
                            }       }     } }
        System.out.println("Input given:"+input);
        if(output != null)
        {
```

```
                System.out.println("Data found in HashTable:"+output);
        }
        else
        {
                System.out.println("Data not found in HashTable");
        }       }
private Hashtable<String, String>retrieveData(Connection conn, String query)
{
        Hashtable<String, String>hashData = newHashtable<String, String>();
        try
        {
                PreparedStatementpstmt = conn.prepareStatement(query);
                ResultSetrs = pstmt.executeQuery();
                while(rs.next())
                 {
                        hashData.put(rs.getString("st_name"), rs.getString("st_mobile"));
                }
        catch (SQLExceptione)
        {
                e.printStackTrace();
        }
    return hashData;
} }
```

Output:-



13. Write a java program that takes tab separated data (one record per line) from a text file and inserts them into a database.

<u>Program:-</u>

Prog11.txt

JNTU 65656565
OU 64646464

ConnectionUtil.Java

```
package LabProgs;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
public class ConnectionUtil
{

	public static Connection getConnection() throws SQLException
{
		Connection connection = null;
		try {
			Class.forName("com.mysql.jdbc.Connection");
			connection =
DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/test", "root", "system");
		} catch (ClassNotFoundException e)
{
			e.printStackTrace();
		}
catch (SQLException e) {
			e.printStackTrace();
		}
		return connection;
	}
}
```

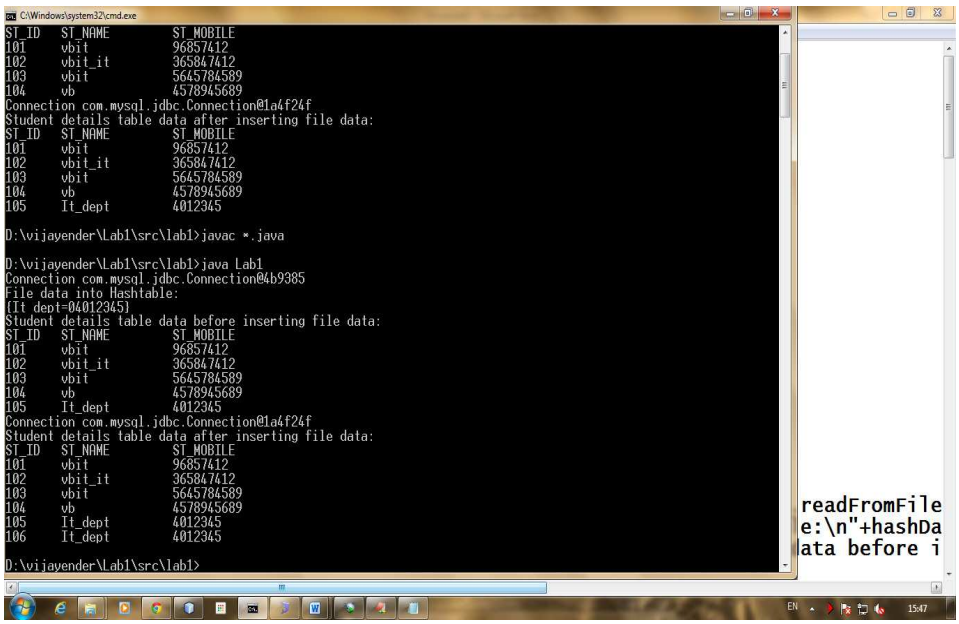
Prog13.Java

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Hashtable;
import java.util.Iterator;

public class Lab1 {

public static void main(String[] args) {
Connection conn = null;
try {
conn = ConnectionUtil.getConnection();
```

```
Lab1 prog13 = new Lab1();
Hashtable<String, String>hashData = prog13.readFromFile("prog11.txt");
System.out.println("File data into Hashtable:\n"+hashData);
System.out.println("Student details table data before inserting file data:");
prog13.retrieveData(conn);
prog13.writeDataToDatabase(conn, hashData);
System.out.println("Student details table data after inserting file data:");
prog13.retrieveData(conn);
} catch (SQLException e) {
e.printStackTrace();
} finally {
try {
conn.close();
} catch (SQLException e) {
e.printStackTrace();
}
}
}

private void writeDataToDatabase(Connection conn,
Hashtable<String, String>hashData) {
String query = "insert into student_details values (?,?,?)";
try {
PreparedStatement pstmt = conn.prepareStatement(query);
if(hashData != null) {
Iterator<String> iterator = hashData.keySet().iterator();
while(iterator.hasNext()) {
String key =  iterator.next();
String value = hashData.get(key);
long id = getNextId("student_details");
pstmt.setString(1,key);
pstmt.setLong(2, new Long(value));
pstmt.setLong(3, id);
pstmt.executeUpdate();
}
}
} catch (SQLException e) {
e.printStackTrace();
}
}

private long getNextId(String tableName) {
String query = "select max(st_id) from " + tableName;
Connection conn;
long id = -1;
try {
conn = ConnectionUtil.getConnection();
Statement statement = conn.createStatement();
ResultSet resultSet = statement.executeQuery(query);
resultSet.next();
id =resultSet.getLong(1);
} catch (SQLException e) {
```

```
e.printStackTrace();
}
return ++id;
}

private void retrieveData(Connection conn) {
try {
String query = "select * from student_details";
PreparedStatement pstmt = conn.prepareStatement(query);
ResultSet rs = pstmt.executeQuery();
System.out.println("ST_ID\tST_NAME\t\tST_MOBILE");
while(rs.next()) {
System.out.println(rs.getString("st_id")+"\t"+rs.getString("st_name")+"\t\t"+rs.getString("st_mo
bile"));
}
} catch (SQLException e) {
e.printStackTrace();
}
}
private Hashtable<String, String>readFromFile(String fileName) {
Hashtable<String, String>hashData = new Hashtable<String, String>();
BufferedReader br = null;
try {
File f = new File("D:\\java//"+fileName);
br = new BufferedReader(new FileReader(f));
String line = null;
while((line = br.readLine()) != null) {
String[] details = line.split("\t");
hashData.put(details[0], details[1]);
}
} catch (FileNotFoundException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
} finally {
try {
br.close();
} catch (IOException e) {
e.printStackTrace();
}
}
return hashData;
}
}
```

Output:-



14. Write a java program that prints the meta-data of a given table.

Program:-

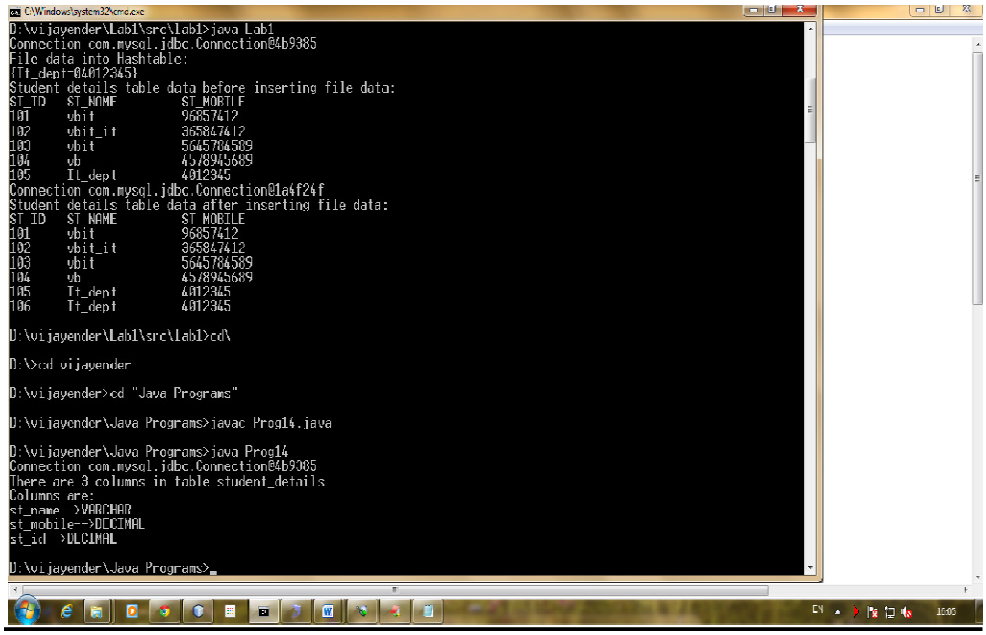
ConnectionUtil.Java

```
package LabProgs;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
public class ConnectionUtil
{
	public static Connection getConnection() throws SQLException {
		Connection connection = null;
		try {
			Class.forName("com.mysql.jdbc.Connection");
		connection = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/test",
		"root", "system");
		} catch (ClassNotFoundException e) {
			e.printStackTrace();
		} catch (SQLException e) {
			e.printStackTrace();
		}
		return connection;
	}


import java.io.BufferedReader;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Hashtable;

import  java.util.Iterator;

import  java.util.logging.Level;

import java.util.logging.Logger;

public class Prog14 {
```

```
        public static void main(String args[]) {
try {
            Connection conn = null;
conn = ConnectionUtil.getConnection();
            Prog14 prog14 = new  Prog14();
prog14.printMetaData(conn, "student_details");
        } catch (SQLException ex) {
Logger.getLogger(Prog14.class.getName()).log(Level.SEVERE, null, ex);
        }
        }

        private void printMetaData(Connection conn, String tableName) {
                String query = "select * from " + tableName;
                Statement statement;
                try {
                        statement = conn.createStatement();
                ResultSetMetaDatametaData = statement.executeQuery(query).getMetaData();
                        longcolSize = metaData.getColumnCount();
                        System.out.println("There are "+ colSize + " columns in table
"+tableName);
                        System.out.println("Columns are:");
                        for(inti=1;i<=colSize;i++) {
                                String colName = metaData.getColumnLabel(i);
                                String colType = metaData.getColumnTypeName(i);
                                System.out.println(colName+"-->"+colType);
                        }
                } catch (SQLException e) {
                        e.printStackTrace();
```

```
                }

        }

}
```

Output:-



Additional Programs:-

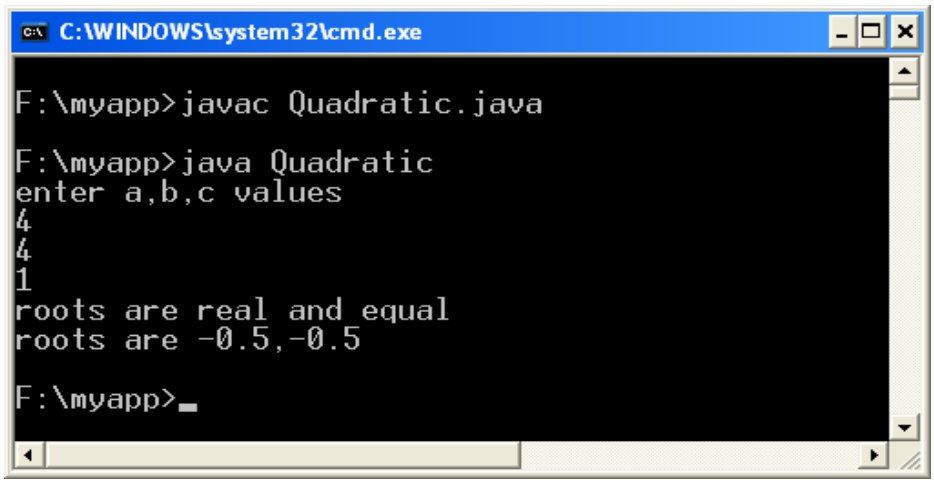
1. Write a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions.

Program :-

```
import java.io.*;

class Quadratic

{

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

{

double x1,x2,disc,a,b,c;

InputStreamReader obj=new InputStreamReader(System.in);

BufferedReader br=new BufferedReader(obj);

System.out.println("enter a,b,c values");

a=Double.parseDouble(br.readLine());

b=Double.parseDouble(br.readLine());

c=Double.parseDouble(br.readLine());

disc=(b*b)-(4*a*c);

if(disc==0)

{

System.out.println("roots are real and equal ");

x1=x2=-b/(2*a);


System.out.println("roots are "+x1+","+x2);

}

else if(disc>0)

{

 System.out.println("roots are real and unequal");

x1=(-b+Math.sqrt(disc))/(2*a);

x2=(-b+Math.sqrt(disc))/(2*a);
```

```
System.out.println("roots are "+x1+","+x2);

}

else

{

 System.out.println("roots are imaginary");

}    } }
```

Input & Output :-

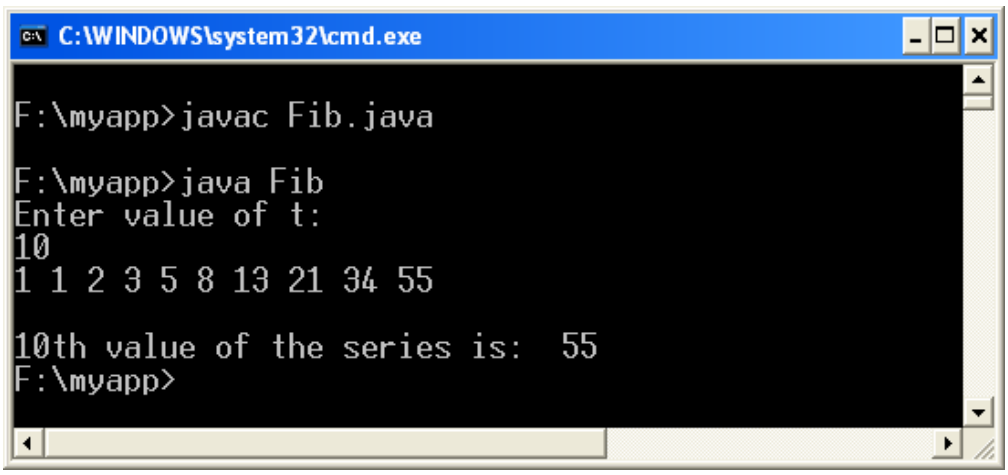


2. The Fibonacci sequence is defined by the following rule. The first 2 values in the sequence are 1, 1. Every subsequent value is the sum of the 2 values preceding it. Write a Java program that uses both recursive and non-recursive functions to print the nth value of the Fibonacci sequence.

<u>Program:-</u>

```
/*Non  Recursive  Solution*/

    import  java.util.Scanner;

    class Fib {

        public static void main(String args[  ])  {

                Scanner input=new Scanner(System.in);

                int i,a=1,b=1,c=0,t;

                System.out.println("Enter value of t:");

                t=input.nextInt();

                System.out.print(a);

                System.out.print(" "+b);

                for(i=0;i<t-2;i++) {

                        c=a+b;

                        a=b;

                        b=c;

                        System.out.print(" "+c);

                }

                System.out.println();

                System.out.print(t+"th value of the series is: "+c);

        }

    }
```

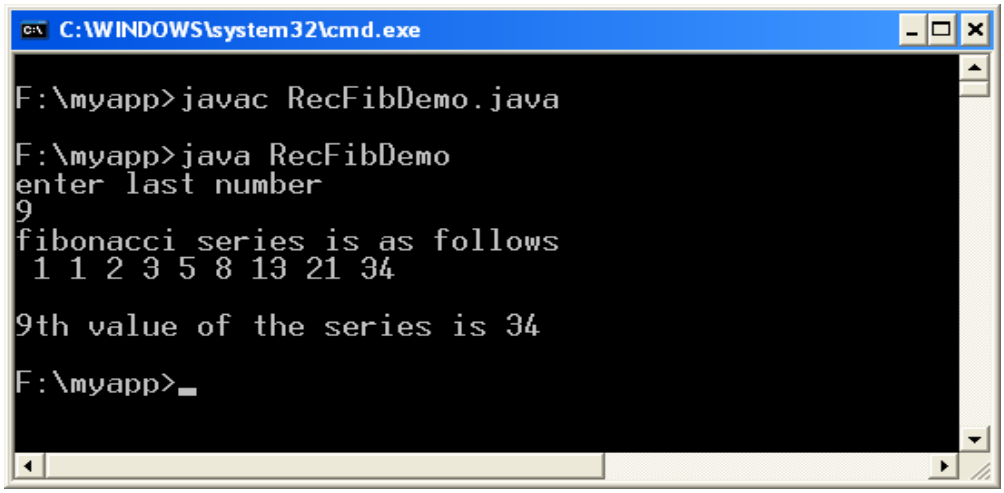
Input & Output:-




```
/* Recursive Solution*/

    import java.io.*;

    import java.lang.*;


    class Demo {

            int fib(int n) {

                    if(n==1)

                        return (1);


                    else if(n==2)

                        return (1);


                    else

                        return (fib(n-1)+fib(n-2));

            }

    }


    class RecFibDemo {
```

```
public static void main(String args[])throws IOException {
        InputStreamReader obj=new InputStreamReader(System.in);
        BufferedReader br=new BufferedReader(obj);
        System.out.println("enter last number");
        int n=Integer.parseInt(br.readLine());
        Demo ob=new Demo();
        System.out.println("fibonacci series is as follows");
        int res=0;
        for(int i=1;i<=n;i++) {
                res=ob.fib(i);
                System.out.println(" "+res); }
        System.out.println();
        System.out.println(n+"th value of the series is "+res);
        } }
```

Input & Output :-

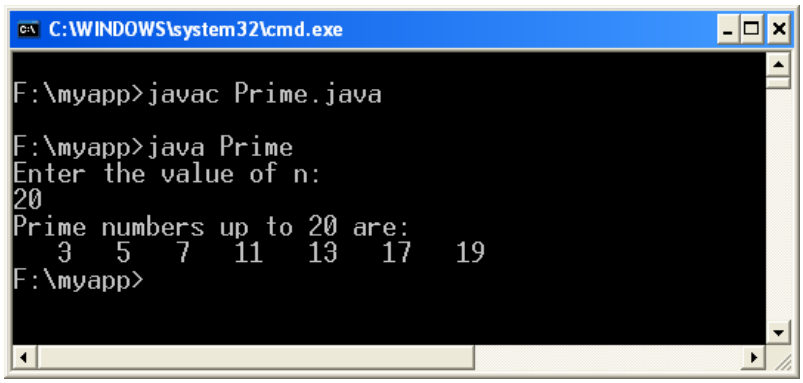


3. WAJP that prompts the user for an integer and then prints out all the prime numbers up to that Integer.

Program:-

```
import java.util.*

class Test {

        void check(int num) {

                System.out.println ("Prime numbers up to "+num+" are:");

                for (int i=1;i<=num;i++)

                        for (int j=2;j<i;j++) {

                            if(i%j==0)

                                break;

                             else if((i%j!=0)&&(j==i-1))

                                  System.out.print(" "+i);

                        }

                }

} //end of class Test

class Prime {

        public static void main(String args[ ]) {

                Test obj1=new Test();

                Scanner input=new Scanner(System.in);

                System.out.println("Enter the value of n:");

                int n=input.nextInt();

                obj1.check(n);

        } }
```

Input & Output :-

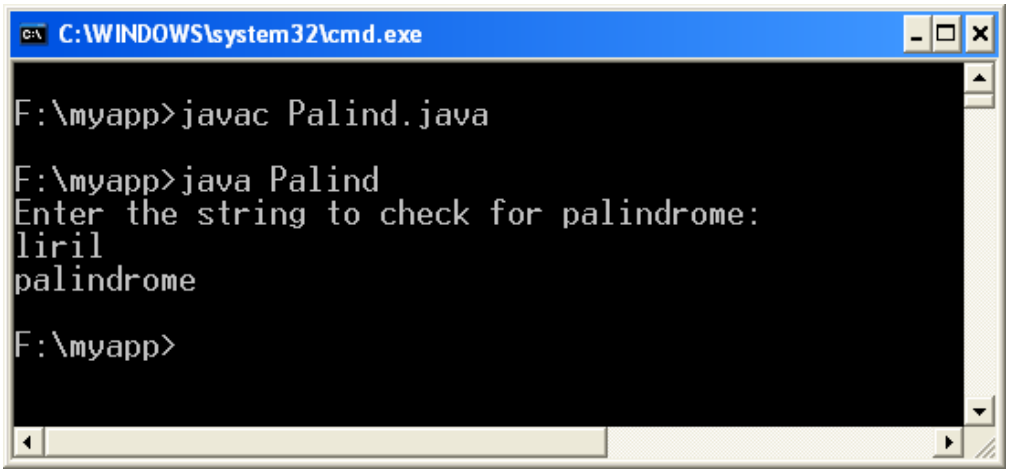


4. WAJP that checks whether a given string is a palindrome or not. Ex: MADAM is a palindrome.

Program:-

```
import java.io.*;

class Palind {

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

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Enter the string to check for palindrome:");

        String  s1=br.readLine();

        StringBuffer sb=new StringBuffer();

        sb.append(s1);

        sb.reverse();

        String s2=sb.toString();

        if(s1.equals(s2))

                System.out.println("palindrome");

        else

                System.out.println("not palindrome");

                } }
```

Input & Output :-



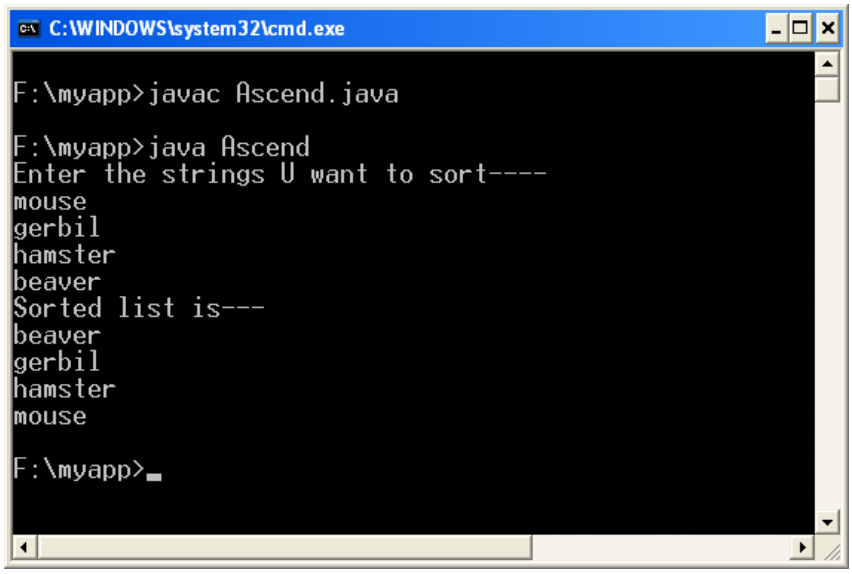
5. WAJP for sorting a given list of names in ascending order.

Program:-

```
import java.io.*;

class Test {

     int len,i,j;

    String arr[  ];


    Test(int n)  {

            len=n;

            arr=new String[n];

     }


     String[ ] getArray()throws IOException {

          BufferedReader br=new BufferedReader (new InputStreamReader(System.in));

          System.out.println ("Enter the strings U want  to sort --- ");

          for (int i=0;i<len;i++)

               arr[i]=br.readLine();

          return arr;

    }


    String[  ] check()throws  ArrayIndexOutOfBoundsException        {

          for (i=0;i<len-1;i++) {

               for(int j=i+1;j<len;j++) {

                    if ((arr[i].compareTo(arr[j]))>0) {

                       String s1=arr[i];

                       arr[i]=arr[j];

                       arr[j]=s1;
```

```
                }
            }
        }
        return arr;
    }
    void display()throws ArrayIndexOutOfBoundsException {
        System.out.println ("Sorted list is---");
        for (i=0;i<len;i++)
            System.out.println(arr[i]);
        }
}     //end of the Test class
class Ascend {
    public static void main(String args[ ])throws IOException {
        Test obj1=new Test(4);
        obj1.getArray();
        obj1.check();
        obj1.display();
    }
}
```

Input & Output :-



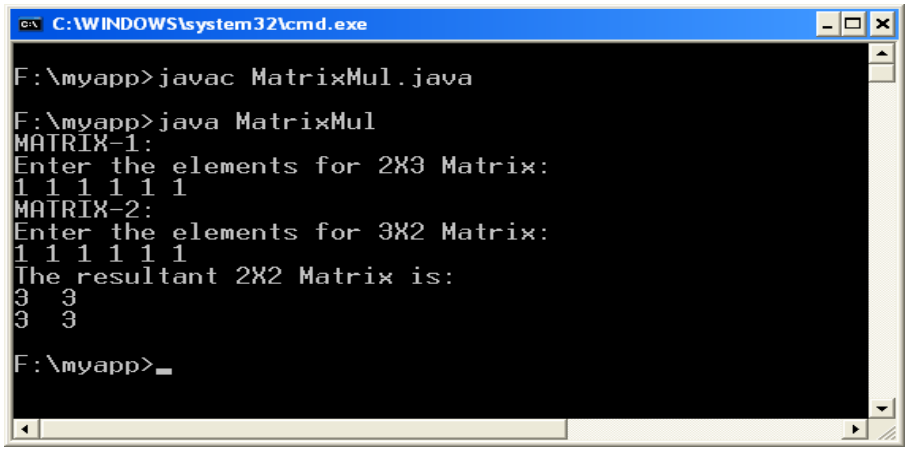
6. WAJP to multiply two given matrices.

Program :-

```
import java.util.*;
class Test {
        int r1,c1,r2,c2;
Test(int r1,int c1,int r2,int c2) {
                this.r1=r1;
                this.c1=c1;
                this.r2=r2;
                this.c2=c2;
        }
        int[ ][ ] getArray(int r,int c) {
                int arr[][]=new  int[r][c];
                System.out.println("Enter the elements for "+r+"X"+c+" Matrix:");
                Scanner input=new Scanner(System.in);
                for(int i=0;i<r;i++)
                for(int j=0;j<c;j++)
                        arr[i][j]=input.nextInt();
                return arr;
        }
        int[ ][ ] findMul(int a[ ][ ],int b[ ][ ]) {
                int c[][]=new int[r1][c2];
                for (int i=0;i<r1;i++)
                    for (int j=0;j<c2;j++) {
                        c[i][j]=0;
                        for (int k=0;k<r2;k++)
                            c[i][j]=c[i][j]+a[i][k]*b[k][j];
                    }
```

```
            return c;
        }
        void putArray(int res[ ][ ]) {
            System.out.println ("The resultant "+r1+"X"+c2+" Matrix is:");
            for (int i=0;i<r1;i++) {
              for (int j=0;j<c2;j++)
                  System.out.print(res[i][j]+" ");
              System.out.println();
            } } } //end of Test class
class MatrixMul {
        public static void main(String args[ ])throws IOException {
            Test obj1=new Test(2,3,3,2);
            Test obj2=new Test(2,3,3,2);
             int x[ ][ ],y[ ][ ],z[ ][ ];
             System.out.println("MATRIX-1:");
            x=obj1.getArray(2,3);        //to get the matrix from user
            System.out.println("MATRIX-2:");
            y=obj2.getArray(3,2);
            z=obj1.findMul(x,y);     //to perform the multiplication
            obj1.putArray(z);          // to display the resultant  matrix
        } }
```

Input & Output :-



7. WAJP that reads

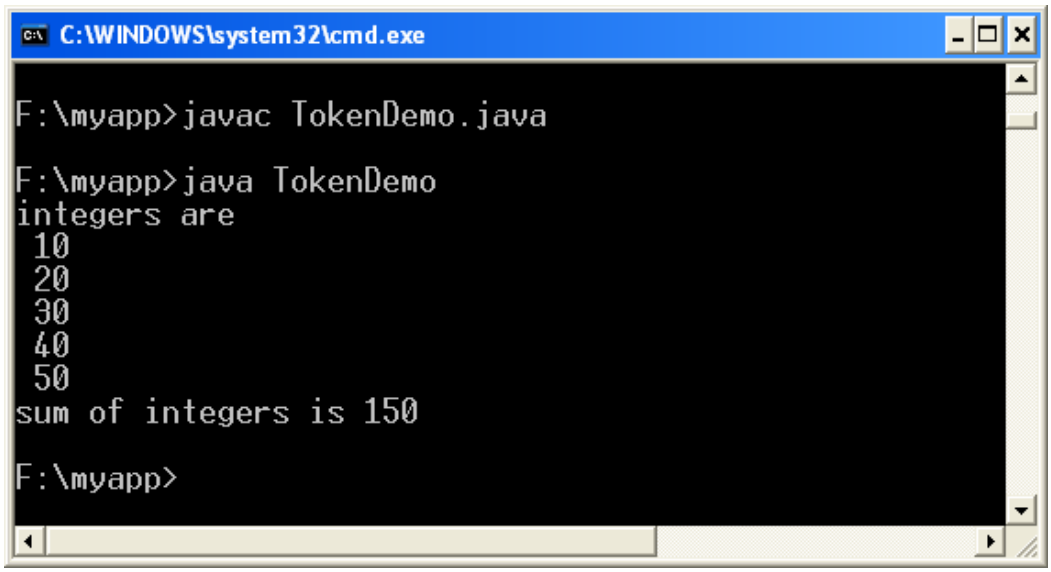
a line of integers and then displays each integer and the sum of all integers. (use StringTokenizer class)

<u>Program :-</u>

```
// Using StringTokenizer class

import java.lang.*;

import java.util.*;

class tokendemo {

    public static void main(String args[ ]) {

        String s="10,20,30,40,50";

        int sum=0;

        StringTokenizer a=new StringTokenizer(s,",",false);

        System.out.println("integers are ");

        while(a.hasMoreTokens()) {

                int b=Integer.parseInt(a.nextToken());

                sum=sum+b;

                System.out.println(" "+b);

        }

        System.out.println("sum of integers is "+sum);

    }

    }

// Alternate solution using command line arguments

class Arguments {

        public static void main(String args[ ]) {

                 int sum=0;

                  int n=args.length;

                 System.out.println("length is "+n);


                int arr[]=new int[n];
```

```
 for(int i=0;i<n;i++)
arr[i]=Integer.parseInt(args[i]);
System.out.println("The enterd values are:");
for(int  i=0;i<n;i++)
System.out.println(arr[i]);
System.out.println("sum of enterd integers is:");
  for(int i=0;i<n;i++)
  sum=sum+arr[i];
System.out.println(sum);
        }
}
```

Input & Output :-

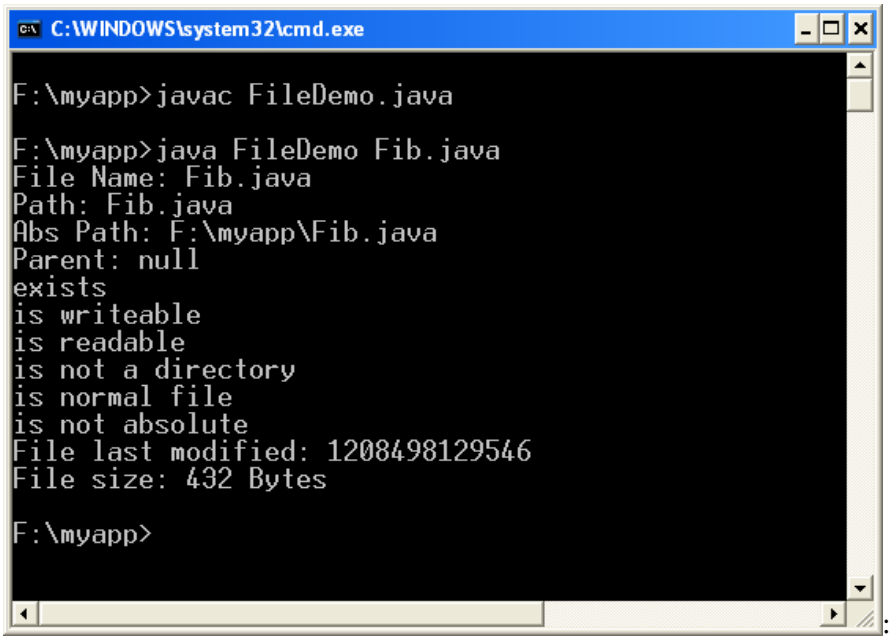


8. WAJP that reads on file name from the user, then displays information about whether the file exists, whether the file is readable, wheteher the file is writable, the type of file and the length of the file in bytes.

Program :-

```
import java.io.File;
class FileDemo {
  static void p(String s) {
    System.out.println(s);
  }

  public static void main(String args[ ]) {
    File f1 = new File(args[0]);
    p("File Name: " + f1.getName());
    p("Path: " + f1.getPath());
    p("Abs Path: " + f1.getAbsolutePath());
    p("Parent: " + f1.getParent());
    p(f1.exists() ? "exists" : "does not exist");
    p(f1.canWrite() ? "is writeable" : "is not writeable");
    p(f1.canRead() ? "is readable" :  "is  not  readable");
    p("is " + (f1.isDirectory() ? "" : "not" + " a directory"));
    p(f1.isFile() ? "is normal file" : "might be a named pipe");
    p(f1.isAbsolute() ? "is absolute" : "is not absolute");
    p("File last modified: " + f1.lastModified());
    p("File size: " + f1.length() + " Bytes");
  } }
```

Input & Output :-

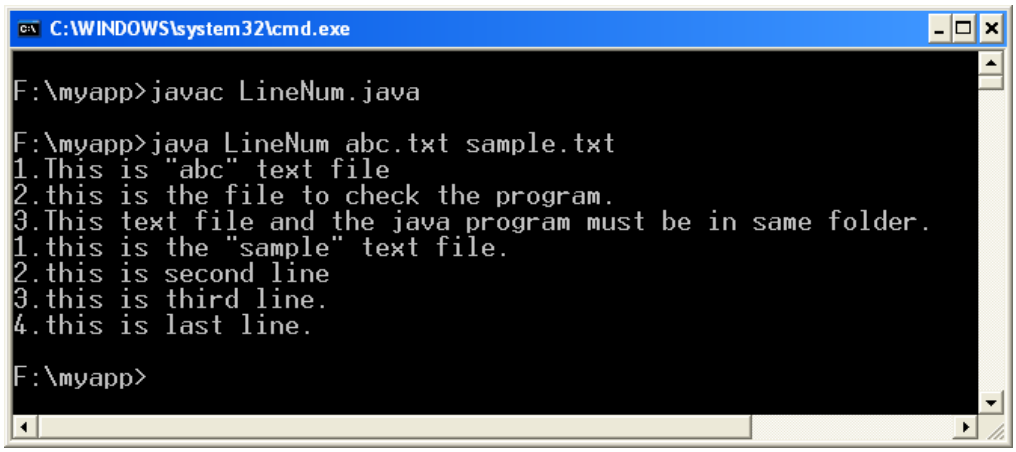


9. WAJP that reads a file and displays the file on the screen, with a line number before each line.

Program :-

```
import java.io.*;

class LineNum{

public static void main(String args[]){

String thisline;

for(int i=0;i<args.length;i++)

{

    try{

            LineNumberReader br=new LineNumberReader(new FileReader(args[i]));

            while((thisline=br.readLine())!=null)

            {

            System.out.println(br.getLineNumber()+"."+thisline);

            }

      }catch(IOException e){

            System.out.println("error:"+e);

      } } } }
```

Input & Output :-



10. WAJP that displays the number of characters, lines and words in a text file.

Program :-

```
import java.io.*;
public class FileStat {
    public static void main(String args[ ])throws IOException {
        long nl=0,nw=0,nc=0;
        String line;
        BufferedReader br=new BufferedReader(new FileReader(args[0]));
        while ((line=br.readLine())!=null) {
            nl++;
            nc=nc+line.length();
            int i=0;
            boolean pspace=true;
            while (i<line.length()) {
                char c=line.charAt(i++);
                boolean cspace=Character.isWhitespace(c);
                if (pspace&&!cspace)
                    nw++;
                pspace=cspace;
            }
        }
        System.out.println("Number of Characters"+nc);
        System.out.println("Number of Characters"+nw);
        System.out.println("Number of Characters"+nl);
 }}

// Alternate solution using StringTokenizer
import java.io.*;
```

```
import java.util.*;

public class FileStat {

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

        long nl=0,nw=0,nc=0;

        String line;

        BufferedReader br=new BufferedReader(new FileReader(args[0]));

        while ((line=br.readLine())!=null) {

            nl++;

            nc=nc+line.length();

            StringTokenizer st = new StringTokenizer(line);
            nw += st.countTokens();
        }
        System.out.println("Number of Characters"+nc);
        System.out.println("Number of Characters"+nw);

        System.out.println("Number of Characters"+nl);

 }}
```

Input & Output :-