Experiment No: 1

Aim:
a) Write a Java program that prompts the user for an integer and then prints out all the prime numbers up to that Integer
b) Write a java program to find the Fibonacci series using recursive and non-recursive functions

Code:
A)

```
import java.util.*;

class Prime
{
	public static void main(String s[])
	{
	System.out.println("Enter Value : ");
	Scanner sc = new Scanner(System.in);
	int number = sc.nextInt();
	System.out.println("Entered Values are Prime Number");
	int no,i=2;

	for(no=0;no<=number;no++)
	{
		for(i=2;i<no;i++)
		{
			if(no%i==0)
			{
				break;
			}
		}
		if(no==i)
		{
			System.out.print(no+" ");
		}
	}
	}
}
```

B)

```
class Fibonacci1
{
  public static void main(String args[])
  {
    int n1=0,n2=1,n3,i,count=10;
    System.out.print(n1+" "+n2);

    for(i=2;i<count;++i)
    {
      n3=n1+n2;
      System.out.print(" "+n3);
      n1=n2;
      n2=n3;
    }
  }
}

class Fibonacci2
{
  static int n1=0,n2=1,n3=0;
  static void printFibonacci(int count)
  {
    if(count>0)
    {
      n3 = n1 + n2;
      n1 = n2;
      n2 = n3;
      System.out.print(" "+n3);
      printFibonacci(count-1);
    }

  }
    public static void main(String args[])
    {
      int count=10;
      System.out.print(n1+" "+n2);
      printFibonacci(count-2);
    }
}
```

Screenshot: [If Applicable]

Conclusion:
Logic are same as C. This was the first task so we have to implement the same logic of C in Java.

Experiment No: 2

Aim:
a) Write a java program for Method overloading and Constructor overloading
b) Write a java program to display the employee details using Scanner class.
c) Write a java program that checks whether a given string is palindrome or not.

Code:

```
/*
Lab Experiment - 02

A) Write a java program for Method overloading and Constructor overloading

Shashank Bagda
*/
class T1
{
	int num1;
	int num2;

	public void add(int i,int j)
	{
		System.out.println(i+j);
	}

	public void add(int i,int j,int k)
	{
		System.out.println(i+j+k);
	}

	T1()
	{
		num1=5;
		num2=10;
	}

	T1(int a,int b)
	{
		num1=a;
		num2=b;
	}

	void Display()
	{
		System.out.println(num1);
		System.out.print(num2);
	}

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

```
                T1 t0=new T1();
                T1 t2=new T1(1,3);
                t2.Display();
                System.out.println("");
                t0.Display();
                System.out.println("");
                t0.add(1,9);
                t2.add(1,6,7);
        }
}





/*
Lab Experiment - 02

B) Write a java program to display the employee details using Scanner class.

Shashank Bagda
*/
import java.util.*;

class Detail
{
        public static void main(String[]args)
                {
                        int emp_id;
                        Scanner s=new Scanner(System.in);
                        System.out.println("Enter employee id ");

                        emp_id=s.nextInt();
                        s.nextLine();
                        String emp_name=new String();
                        Scanner sc=new Scanner(System.in);
                        System.out.println("Enter employee name ");

                        emp_name=s.nextLine();

                        System.out.println("Employee id:"+emp_id);

                        System.out.println("Employee name :"+emp_name);

                }
}
```

```
/*
Lab Experiment - 02

c) Write a java program that checks whether a given string is palindrome or not.

Shashank Bagda
*/

import java.util.*;

class Palindrome
{
        public static void main(String[]args)
        {
                String first=new String();
                String second=new String();
                Scanner sc=new Scanner(System.in);

                System.out.println("Enter the string ");

                first=sc.nextLine();

                for(int i=first.length()-1;i>=0;i--)
                {
                        second=second+first.charAt(i);
                }

                if(first.equalsIgnoreCase(second))
                {
                        System.out.println("String  is Palindrome");
                }

                else
                {
                        System.out.println("String is not Palindrome");
                }
        }
}
```

Screenshot: [If Applicable]

Conclusion:
From the above tasks we came to know and use of methods like overloading and overriding. Also we did some tasks like palindrome and using of scanner class.

Experiment No: 3

Aim:
a) Write a java program to represent Abstract class with example
b) Write a java program to implement Interface using extends keyword.

Code:

```
/*
Lab Experiment - 03
Shashank Bagda
*/

// (A) Represent Abstract Class

abstract class A
{
  abstract void run();
}

// (B) Implement Interface using extends

class B extends A
{
  void run()
  {
    System.out.println("Lab Experiment 03");
  }

  public static void main(String args[])
  {
    A obj = new B();
    obj.run();
  }
}
```

Screenshot: [If Applicable]

Conclusion:
After completion of the following task we came to know the difference between the abstract class and the interface. We also came to know the use of both the methods.

Experiment No: 4

Aim:
Write a java program to create user defined package. Import Package and Use it functionality in another Java File.

Code:

```
package TestPackage;

public class Mypack
{
   public void display()
       {
          System.out.println("User Defined Package");
       }
}
```

```
import TestPackage.*;

class Main extends  Mypack
{
            public static void main(String s[])
            {
                        Mypack t1 = new Mypack();
                        t1.display();
            }
}
```

Screenshot: [If Applicable]

Conclusion:
By doing the task we are able to create package i.e. a folder for class files. In the industrial project package plays an important role and after this we are able to implement it in our basic tasks.

Experiment No: 5

Aim:
a) Write a Java Program to demonstrate Exception Handling using try, catch and finally.
b) Write a Custom Exception and Use it.

Code:

A)

```
/*
Lab Experiment 5
a) Write a Java Program to demonstrate Exception Handling using try, catch and finally.

Shashank Bagda - 20
*/
class Main
{
  public static void main (String args[])
  {
    //Try Block
    try
    {
      System.out.println ("Try Block");
      int div = 125 / 5;
      System.out.println ("Answer : " + div);
    }

    //Catch Block
    catch (NullPointerException e)
    {
      System.out.println ("Catch Block");
      System.out.println (e);
    }

    //Finally Block
    finally
    {
      System.out.println ("Finally Block");
    }

  }
}
```

B)

```
/*
Lab Experiment 05
b) Write a Custom Exception and Use it.

Shashank Bagda - 20
*/

class MyException extends Exception
{

}

// A Class that uses above MyException
public class Test
{
   public static void main(String args[])
   {
      try
      {
         // Throw an object of user defined exception
         throw new MyException();
      }
      catch (MyException ex)
      {
         System.out.println("Caught");
         System.out.println(ex.getMessage());
      }
   }
}
```

Screenshot: [If Applicable]

Conclusion:
There are so many methods and algorithms which throws error and also exception cases. To handle the situation and to run the code smoothly we have to use try and catch.

Experiment No: 6

Aim:
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.

Code:

```
import java.util.Random;

class RandomNumberThread extends Thread
{
   public void run()
   {
      Random random = new Random();

      while(true)
      {
         int randomInteger = random.nextInt(100);

         System.out.println("Random Integer generated : " + randomInteger);

         if((randomInteger%2) == 0)
         {
            SquareThread sThread = new SquareThread(randomInteger);
            sThread.start();
         }
         else
         {
            CubeThread cThread = new CubeThread(randomInteger);
            cThread.start();
         }

         try
         {
            Thread.sleep(1000);
         }
         catch (InterruptedException ex)
         {
            System.out.println(ex);
         }
      }
   }
}

class SquareThread extends Thread
{
   int number;
```

```
    SquareThread(int randomNumbern)
    {
        number = randomNumbern;
    }

    public void run()
    {
        System.out.println("Square of " + number + " = " + (number * number));
        System.out.println("----------------------------");
    }
}

class CubeThread extends Thread
{
    int number;

    CubeThread(int randomNumber)
    {
        number = randomNumber;
    }

    public void run()
    {
        System.out.println("Cube of " + number + " = " + number * number * number);
    }
}

public class Test
{
    public static void main(String args[])
    {
        RandomNumberThread rnThread = new RandomNumberThread();
        rnThread.start();
    }
}
```

Screenshot: [If Applicable]

Conclusion:
Multi-Threading is very essential for the controlling of the application development and also those codes who requires a control on execution. By completion of the following task we are able to implement that utility of java in our basic tasks also.

Experiment No: 7

Aim:
Write a Java program to implement Collection Framework.

Code:

```
import java.util.*;
class Main
{
 public static void main(String args[])
  {
  ArrayList<String> list=new ArrayList<String>();
  list.add("ICT");
  list.add("JAVA");
  list.add("Shashank");
  list.add("Marwadi University");
  Iterator itr=list.iterator();//Created a Iterator
  while(itr.hasNext()) //hasNext is used to check if there is any element present or not
    {
       System.out.println(itr.next());//this will print the element and move to next line
    }
   }
}
```

Screenshot: [If Applicable]

Conclusion:
In this task we are introduced with a new method called Iterator. We can add arraylist and with the help of iterator we are able to print them in a sequence.

Experiment No: 8
Aim: Write a Java program that reads a file and displays the file on the screen, with a line number before each line.
Code:
Screenshot: [If Applicable]
Conclusion:

Experiment No: 9
Aim: 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)
Code:
Screenshot: [If Applicable]
Conclusion:

Experiment No: 10

Aim:
Develop an AWT/SWING program that receives an integer in one text field & compute its factorial value & returns it in another text filed when the button “Compute” is clicked

Code:

ACTION LISTENER :

```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.*;
import java.lang.*;
import java.util.*;
public class MyActionListener implements ActionListener
{
        MyFrame mf;
        String s1,s2;
        MyActionListener(MyFrame m)
        {
                this.mf=m;
        }
        public void actionPerformed(ActionEvent e)
        {
                int value=1;
                s1= this.mf.t1.getText();
                int i = Integer.parseInt(s1);
                for(int j=1;j<=i;j++)
                {
                        value=value*j;
                }
                this.mf.t2.setText(new Integer(value).toString());
                System.out.println(value);
        }
}
```

MY FRAME :

```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.*;
import java.lang.*;

public class MyFrame extends Frame
{
        Button b1;
        TextField t1,t2;
```

```
	Font f1;
	MyActionListener al = new MyActionListener(this);
	MyFrame()
	{
		super("My Test Application");
		setLayout(null);
		setBounds(0,0,800,800);

		f1 = new Font("Arial",Font.ITALIC,30);
		t1=new TextField();
		t2=new TextField();
		b1 = new Button("COMPUTE");

		t1.setFont(f1);
		t2.setFont(f1);
		b1.setFont(f1);

		add(b1);
		add(t1);
		add(t2);

		t1.setBounds(20,150,400,100);
		b1.setBounds(20,400,400,100);
		t2.setBounds(20,600,400,100);

		b1.addActionListener(al);

		addWindowListener(new WindowAdapter()
			{
				public void windowClosing(WindowEvent we)
				{
					System.exit(0);
				}
			});

	}

	public static void main(String s[]) throws IOException
	{
		MyFrame mf = new MyFrame();
		mf.setVisible(true);
	}
}
```

Screenshot: [If Applicable]

Conclusion:
We are able to perform the factorial function by using the MVC

Experiment No: 11

Aim:
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.

Code:

MYFRAME :

```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.*;
import java.lang.*;

public class MyFrame extends Frame
{
        TextField txt_Scr;
        Button b0,b1,b2,b3,b4,b5,b6,b7,b8,b9;
        Button plus,minus,multi,divide,clr,ans;

        MyActionListener  m1 = new MyActionListener(this);
         MyFrame()
        {
                super("Calculator");
                setLayout(null);
                setBounds(200,200,400,540);

                txt_Scr = new TextField();
                txt_Scr.setBounds(20,30,360,100);
                b0 = new Button("0");
                b1 = new Button("1");
                b2 = new Button("2");
                b3 = new Button("3");
                b4 = new Button("4");
                b5 = new Button("5");
                b6 = new Button("6");
                b7 = new Button("7");
                b8 = new Button("8");
                b9 = new Button("9");
                plus = new Button("+");
                minus = new Button("-");
                multi= new Button("*");
                divide= new Button("/");
                clr = new Button("CE");
                ans = new Button("=");
                add(txt_Scr);
                add(b0);
                add(b1);
                add(b2);
                add(b3);
                add(b4);
                add(b5);
                add(b6);
                add(b7);
                add(b8);
                add(b9);
                add(plus);
                add(minus);
                add(multi);
                add(divide);
                add(clr);
                add(ans);

                b0.setBounds(0,440,100,100);
                b1.setBounds(0,140,100,100);
                b2.setBounds(100,140,100,100);
                b3.setBounds(200,140,100,100);
                b4.setBounds(0,240,100,100);
                b5.setBounds(100,240,100,100);
                b6.setBounds(200,240,100,100);
                b7.setBounds(0,340,100,100);
                b8.setBounds(100,340,100,100);
                b9.setBounds(200,340,100,100);
                plus.setBounds(200,440,100,100);
                minus.setBounds(300,440,100,100);
                multi.setBounds(300,340,100,100);
                divide.setBounds(300,240,100,100);
                clr.setBounds(300,140,100,100);
                ans.setBounds(100,440,100,100);


                txt_Scr.addActionListener(m1);
                b0.addActionListener(m1);
                b1.addActionListener(m1);
                b2.addActionListener(m1);
                b3.addActionListener(m1);
                b4.addActionListener(m1);
                b5.addActionListener(m1);
                b6.addActionListener(m1);
                b7.addActionListener(m1);
                b8.addActionListener(m1);
                b9.addActionListener(m1);
                plus.addActionListener(m1);
                minus.addActionListener(m1);
                multi.addActionListener(m1);
```

```
		divide.addActionListener(m1);
		clr.addActionListener(m1);
		ans.addActionListener(m1);

		Font f1 = new Font ("Arial",Font.BOLD,35);
		txt_Scr.setFont(f1);
		b0.setFont(f1);
		b1.setFont(f1);
		b2.setFont(f1);
		b3.setFont(f1);
		b4.setFont(f1);
		b5.setFont(f1);
		b6.setFont(f1);
		b7.setFont(f1);
		b8.setFont(f1);
		b9.setFont(f1);
		plus.setFont(f1);
		minus.setFont(f1);
		multi.setFont(f1);
		divide.setFont(f1);
		clr.setFont(f1);
		ans.setFont(f1);

			addWindowListener(new WindowAdapter()
			{
				public void windowClosing(WindowEvent we)
				{
					System.exit(0);
				}
			});

	}
	public static void main(String args[])
	{
			 MyFrame mf = new  MyFrame();
			mf.setVisible(true);
	}

}
```

ACTIONLISTENER :

```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.*;
import java.lang.*;
```

```
public class MyActionListener implements ActionListener
{
        MyFrame mf;
        String no1 = "",no2 = "";
        int i = 0;
        MyActionListener(MyFrame m)
        {
                this.mf = m;
        }
        public void actionPerformed(ActionEvent e)
        {
                if(e.getSource().equals(this.mf.clr))
                {
                        this.mf.txt_Scr.setText("");
                }
                else if(e.getSource().equals(this.mf.plus))
                {
                        no1 = this.mf.txt_Scr.getText();
                        this.mf.txt_Scr.setText("");
                        i = i + 1;
                }
                else if(e.getSource().equals(this.mf.minus))
                {
                        no1 = this.mf.txt_Scr.getText();
                        this.mf.txt_Scr.setText("");
                        i = i + 2;
                }
                else if(e.getSource().equals(this.mf.multi))
                {
                        no1 = this.mf.txt_Scr.getText();
                        this.mf.txt_Scr.setText("");
                        i = i + 3;
                }
                else if(e.getSource().equals(this.mf.divide))
                {
                        no1 = this.mf.txt_Scr.getText();
                        this.mf.txt_Scr.setText("");
                        i = i + 4;
                }
                else if(e.getSource().equals(this.mf.ans))
                {
                        no2 = this.mf.txt_Scr.getText();

                        switch(i)
                        {
                                case 1:
                                {
                                        this.mf.txt_Scr.setText(new String(new
Integer(Integer.parseInt(no1)+Integer.parseInt(no2)).toString()));
                                        i = 0;
                                        break;
                                }
                                case 2:
                                {
                                        this.mf.txt_Scr.setText(new String(new
Integer(Integer.parseInt(no1)-Integer.parseInt(no2)).toString()));
                                        i = 0;
                                        break;
                                }
                                case 3:
                                {
                                        this.mf.txt_Scr.setText(new String(new
Integer(Integer.parseInt(no1)*Integer.parseInt(no2)).toString()));
                                        i = 0;
                                        break;
                                }
                                case 4:
                                {
                                        this.mf.txt_Scr.setText(new String(new
Float(Float.parseFloat(no1)/Float.parseFloat(no2)).toString()));
                                        i = 0;
                                        break;
                                }
                        }
                }
                else
                {

        this.mf.txt_Scr.setText((this.mf.txt_Scr.getText()+e.getActionCommand()).toString
());
                }

        }
}
```

Screenshot: [If Applicable]

Conclusion:
By this task we are able to create a GUI based calculator which performs the basic arithmetic operation like +, - , / , * .

Experiment No: 12

Aim:
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

Code:

MYFRAME :

```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.*;
import java.lang.*;

public class MyFrame extends Frame
{
        TextField t1,t2,t3;
        Label l1,l2,l3;
        Button b1;
        Font f1;

        MyActionListener ml = new MyActionListener(this);
        MyFrame()
        {
                super("Division Exception");
                setLayout(null);
                setBounds(0,0,1600,1000);

                t1 = new TextField();
                t2 = new TextField();
                t3 = new TextField();

                b1 = new Button("DIVIDE");

                l1 = new Label("NUMBER 1");
                l2 = new Label("NUMBER 2");
                l3 = new Label(" ANSWER ");

                //Set Component
                l1.setBounds(50,50,425,200);
                t1.setBounds(525,50,425,200);

                l2.setBounds(50,300,425,200);
                t2.setBounds(525,300,425,200);
```

```
		l3.setBounds(50,550,425,200);
		t3.setBounds(525,550,425,200);

		b1.setBounds(1000,175,425,200);

		Font f1 = new Font("Arial",Font.BOLD,30);

		//SetFont
		l1.setFont(f1);
		l2.setFont(f1);
		l3.setFont(f1);
		t1.setFont(f1);
		t2.setFont(f1);
		t3.setFont(f1);
		b1.setFont(f1);

		//Add Component to Frame
		add(t1);
		add(t2);
		add(t3);
		add(l1);
		add(l2);
		add(l3);
		add(b1);

		b1.addActionListener(ml);

		addWindowListener(new WindowAdapter()
			{
				public void windowClosing(WindowEvent we)
				{
					System.exit(0);
				}
			});
	}
	public static void main(String args[])
	{
			MyFrame mf = new MyFrame();
			mf.setVisible(true);
	}
}
```

ACTIONLISTENER :

```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.*;
import java.lang.*;

public class MyActionListener implements ActionListener
{
        MyFrame fm;

        MyActionListener(MyFrame m)
        {
                this.fm = m;
        }

        public void actionPerformed(ActionEvent e)
        {
                System.out.println(e.getActionCommand());

                int value1 = Integer.parseInt(this.fm.t1.getText());
                int value2 = Integer.parseInt(this.fm.t2.getText());

                if(e.getActionCommand().equals("DIVIDE"))
                        {
                                float value;

                                try
                          {
                            value=value1/value2;
                            this.fm.t3.setText(new Float(value).toString());
                                        System.out.println(value);
                          }

                          catch(ArithmeticException ef)
                          {
                            System.out.println(ef);
                            this.fm.t3.setText("ArithmeticException");
                          }

                          catch(NumberFormatException ef)
                          {
                            System.out.println(ef);
                            this.fm.t3.setText("NumberFormatException");
                          }
                            }

        }
}
```

Screenshot: [If Applicable]

Conclusion:
By this task we are able handle the division exception like if we divide any number with 0 the we can exception.

Experiment No: 13

Aim:
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.

Code:

MYFRAME :

```
import java.awt.*;
import java.awt.event.*;
import java.awt.AWTEvent.*;
import java.lang.*;
import javax.swing.*;
public class MyFrame extends JFrame
{

        JFrame f;

        JRadioButton btRed,btYellow,btGreen;

        JTextField fieldBox;

        JButton clear;

        MyActionListener ml = new MyActionListener(this);

        MyFrame()
        {

                f=new JFrame("Traffic Light");
                f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

                btRed = new JRadioButton("Red light");
                btYellow = new JRadioButton("Yellow light");
                btGreen = new JRadioButton("Green light");

                fieldBox = new JTextField(" ");
                fieldBox.setBackground(Color.WHITE);

                clear = new JButton("CLEAR");

                //Set Component

                fieldBox.setBounds(190,130,125,30);

                btRed.setBounds(50,180,125,50);
                btYellow.setBounds(190,180,125,50);
                btGreen.setBounds(330,180,125,50);

                clear.setBounds(150,380,200,50);

                Font f1 = new Font("Arial",Font.BOLD,15);

                //SetFont

                clear.setFont(f1);
                btRed.setFont(f1);
                btYellow.setFont(f1);
                btGreen.setFont(f1);
                fieldBox.setFont(f1);


                f.add(fieldBox);
                f.add(btRed);
                f.add(btYellow);
                f.add(btGreen);
                f.add(clear);

                this.btRed.addActionListener(ml);
                this.btYellow.addActionListener(ml);
                this.btGreen.addActionListener(ml);
                this.clear.addActionListener(ml);
                this.fieldBox.addActionListener(ml);


                f.setLayout(null);
                f.setSize(510,500);
                f.setVisible(true);
        }

        public static void main(String[] args)
        {
                new MyFrame();

        }
}
```

ACTIONLISTENER :

```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.*;
import java.lang.*;
import java.util.*;
```

```
public class MyActionListener implements ActionListener
{
        MyFrame mf;
        MyActionListener(MyFrame f)
        {
                this.mf=f;
        }

        public void actionPerformed(ActionEvent e)
        {
                if(e.getActionCommand().equals("Red light"))
                {
                        this.mf.fieldBox.setText("STOP");
                        this.mf.fieldBox.setBackground(Color.RED);
                }
                else if(e.getActionCommand().equals("Yellow light"))
                {
                        this.mf.fieldBox.setText("READY");
                        this.mf.fieldBox.setBackground(Color.YELLOW);
                }
                else if(e.getActionCommand().equals("Green light"))
                {
                        this.mf.fieldBox.setText("GO");
                        this.mf.fieldBox.setBackground(Color.GREEN);
                }
                else if(e.getActionCommand().equals("CLEAR"))
                {
                        this.mf.fieldBox.setText(null);
                        this.mf.fieldBox.setBackground(Color.WHITE);
                }
        }
}
```

Screenshot: [If Applicable]

Conclusion:
We are able to make a GUI based traffic light function which indicates the respective colour like we the signal is stop, it will indicate red colour and so on.

Experiment No: 14

Aim:
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.

Code:

MYFRAME :

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.lang.*;

public class MyFrame extends Frame
{
        int row=9,column=3;
        Label[][] box = new Label [row][column];
        JButton Fetch;
        Font f1;
        MyActionListener ml = new MyActionListener(this);
        MyFrame()
        {
                setLayout(null);
                this.setBounds(0,0, 610, 750);

                JPanel panel =new JPanel();

                panel.setLayout(new GridLayout(row,column));

                f1 = new Font("Arial",Font.BOLD,30);
                Fetch = new JButton("Fetch");
                Fetch.setFont(f1);
                Fetch.setBounds(200,680,200,50);

                for(int loop1=0;loop1<row;loop1++)
                {
                        for(int loop2=0;loop2<column;loop2++)
                        {
                                box[loop1][loop2] = new Label();
                                box[loop1][loop2].setFont(f1);
                                panel.add(box[loop1][loop2]);
                        }
                }
                Fetch.addActionListener(ml);
                panel.setBounds(10,50,580,600);
                add(Fetch);
                add(panel);
```

```
		addWindowListener(new WindowAdapter()
			{
				public void windowClosing(WindowEvent we)
				{
					System.exit(0);
				}
			});
	}
	public static void main(String s[])
	{

		MyFrame mf = new MyFrame();
		mf.setVisible(true);
	}
}
```

ACTIONLISTENER :

```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.*;
import java.lang.*;
import java.util.*;
import java.io.*;
public class MyActionListener implements ActionListener
{
	MyFrame mf;
	MyActionListener(MyFrame f)
	{
		this.mf=f;
	}
	public void actionPerformed(ActionEvent e)
	{
		if(e.getActionCommand().equals("Fetch"))
		{

			try
			{
				char ch[] = new char[3000];
				FileReader fr = new FileReader("Table.txt");
				fr.read(ch);
				fr.close();

				String records = new String(ch);
				String record[] = records.split("\n");
```

```
                                        for(int loop=0;loop<record.length;loop++)
                                        {
                                                String attr[] = record[loop].split(",");
                                                for(int i=0;i<3;i++)
                                                {
                                                        if(attr.length>1)
                                                        {
                                                                if(i==0)
                                                                {

            this.mf.box[loop][0].setText(attr[0]);
                                                                }
                                                                else if(i==1)
                                                                {

            this.mf.box[loop][1].setText(attr[1]);
                                                                }
                                                                else if(i==2)
                                                                {

            this.mf.box[loop][2].setText(attr[2]);
                                                                }
                                                        }
                                                }
                                        }

                                }
                                catch (IOException ioe)
                                {
                                        System.out.println(ioe);
                                }

                        }

            }
}
```

Screenshot: [If Applicable]

Conclusion:
With this task we are able to read data from a file and print it in a tabular form. This type of program is helpful in codes we have store or present any data in a well mannered form.