



KG REDDY
College of Engineering
& Technology
Engineering India's Changemakers

KG REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Approved by AICTE, New Delhi, Affiliated to JNTUH, Hyderabad)
Chilkur (Village), Moinabad (Mandal), R. R Dist, TS-501504



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING – DATA SCIENCE



JAVA PROGRAMMING LAB MANUAL

Subject Code : KG21CS407

Regulation : KG 21

Academic Year : 2023-2024

II B. TECH II SEMESTER

**COMPUTER SCIENCE AND ENGINEERING – DATA SCIENCE
KG REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY**

Affiliated to JNTUH, Chilkur,(V), Moinabad(M) R. R Dist, TS-501504

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING – DATA SCIENCE

VISION AND MISSION OF THE INSTITUTION

VISION:

To become self-sustainable institution this is recognized for its new age engineering through innovative teaching and learning culture, inculcating research and entrepreneurial ecosystem, and sustainable social impact in the community.

MISSION:

- To offer undergraduate and post-graduate programs that is supported through industry relevant curriculum and innovative teaching and learning processes that would help students succeed in their professional careers.
- To provide necessary support structures for students, this will contribute to their personal and professional growth and enable them to become leaders in their respective fields.
- To provide faculty and students with an ecosystem that fosters research and development through strategic partnerships with government organisations and collaboration with industries.
- To contribute to the development of the region by using our technological expertise to work with nearby communities and support them in their social and economic growth.

VISION AND MISSION OF DEPARTMENT

VISION:

To be recognized as a department of excellence by stimulating a learning environment in which students and faculty will thrive and grow to achieve their professional, institutional and societal goals.

MISSION:

- To provide high quality technical education to students that will enable life-long learning and build expertise in advanced technologies in Computer Science and Engineering.
- To promote research and development by providing opportunities to solve complex engineering problems in collaboration with industry and government agencies.
- To encourage professional development of students that will inculcate ethical values and leadership skills while working with the community to address societal issues.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING – DATA SCIENCE

PROGRAM EDUCATIONAL OBJECTIVES (PEOS):

A graduate of the Computer Science and Engineering Program should:

PEO1	Program Educational Objective1: (PEO1) The Graduates will provide solutions to difficult and challenging issues in their profession by applying computer science and engineering theory and principles.
PEO2	Program Educational Objective2 :(PEO2) The Graduates have successful careers in computer science and engineering fields or will be able to successfully pursue advanced degrees.
PEO3	Program Educational Objective3: (PEO3) The Graduates will communicate effectively, work collaboratively and exhibit high levels of Professionalism, moral and ethical responsibility.
PEO4	Program Educational Objective4 :(PEO4) The Graduates will develop the ability to understand and analyse Engineering issues in a broader perspective with ethical responsibility towards sustainable development.

PROGRAM OUTCOMES (POS):

PO1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering Fundamentals and an engineering specialization to the solution of complex engineering problems.
PO2	Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7	Environment and sustainability: Understand the impact of the professional engineering Solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	Individual and team work: Function effectively as an individual, and as a member or leader In diverse teams, and in multi-disciplinary settings.
PO10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	Project management and finance: Demonstrate knowledge and understanding of the Engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES(PSOs):

PSO1	Problem Solving Skills – Graduate will be able to apply computational techniques and software principles to solve complex engineering problems pertaining to software engineering.
PSO2	Professional Skills – Graduate will be able to think critically, communicate effectively, and collaborate in teams through participation in co and extra-curricular activities.
PSO3	Successful Career – Graduates will possess a solid foundation in computer science and engineering that will enable them to grow in their profession and pursue lifelong learning through post-graduation and professional development.

CONTENTS

S.No	Name of the Experiment	Page No.
1	Use Eclipse or Net bean 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	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 divided by zero.	
3	a) Develop an applet in Java that displays a simple message. b) Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named "Compute" is clicked.	
4	Write a Java 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 Num 2 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 Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.	
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	
6	Write a Java program for the following: Create a doubly linked list of elements. Delete a given element from the above list. Display the contents of the list after deletion.	
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 selected color. Initially, there is no message shown.	
8	Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.	
9	Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in Grid Layout.	
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).	
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).	
12	Write a Java program that correctly implements the producer – consumer problem using the concept of interthread communication.	
13	Write a Java program to list all the files in a directory including the files present in all its subdirectories	
14	Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order	
15	Write a Java program that implements Bubble sort algorithm for sorting in descending order and also shows the number of interchanges occurred for the given set of integers.	

Additional Programs

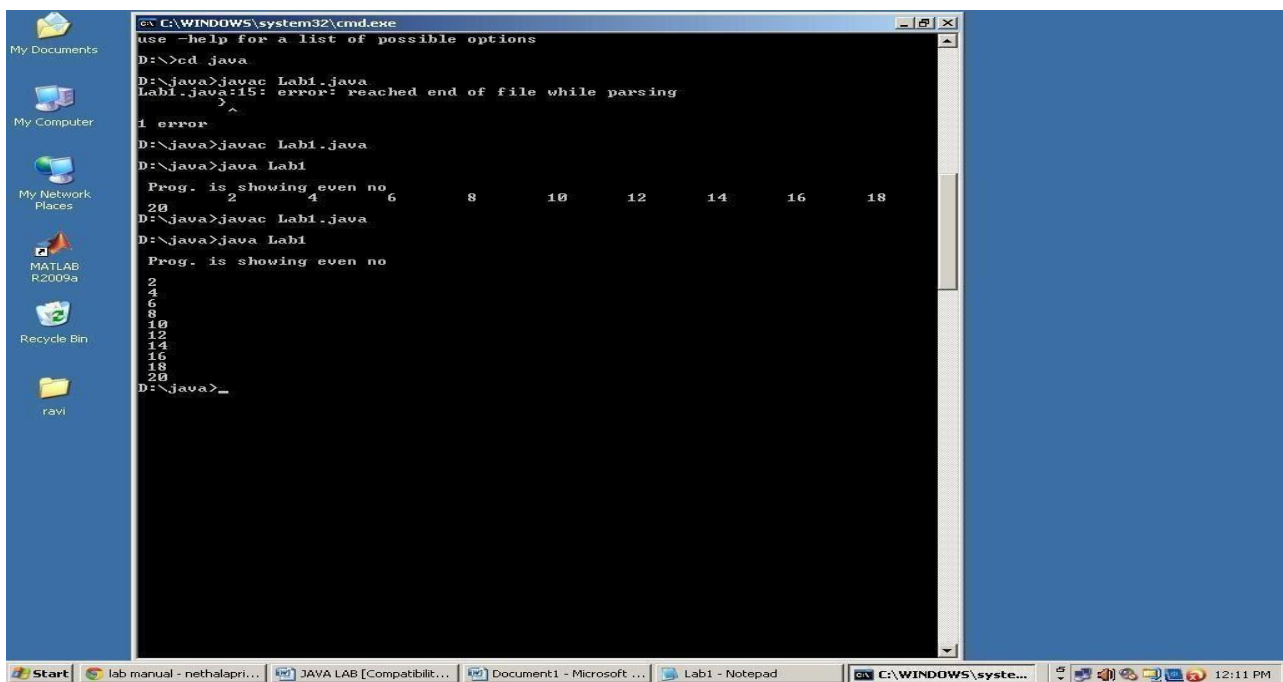
S No	List of the Experiment	Page No
1	Write a java program that connects to a database using JDBC	
2	Write a java program to connect to a database using JDBC and insert values into it	
3	Write a java program to connect to a database using JDBC and delete values from it	
4	Write a java program for handling Mouse events and Key events	

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);
            }
        }
    }
}
```

Output:-



```
C:\WINDOWS\system32\cmd.exe
use -help for a list of possible options
D:\>cd java
D:\java>javac Lab1.java
Lab1.java:15: error: reached end of file while parsing
>
1 error
D:\java>javac Lab1.java
D:\java>java Lab1
Prog. is showing even no
2      4      6      8      10     12     14     16     18
20
D:\java>javac Lab1.java
D:\java>java Lab1
Prog. is showing even no
2
4
6
8
10
12
14
16
18
20
D:\java>_
```

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 java.awt.*;
import java.awt.event.*;
public class Calculator implements ActionListener
{
    int c,n;
    String s1,s2,s3,s4,s5;
    Frame f;
    Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,b16,b17;
    Panel p;
    TextField tf;
    GridLayout g;
    Calculator()
    {
        f = new Frame("My calculator");
        p = new Panel();
        f.setLayout(new FlowLayout());
        b1 = new Button("0");
        b1.addActionListener(this);
        b2 = new Button("1");
        b2.addActionListener(this);
        b3 = new Button("2");
        b3.addActionListener(this);
        b4 = new Button("3");
        b4.addActionListener(this);
        b5 = new Button("4");
        b5.addActionListener(this);
        b6 = new Button("5");
        b6.addActionListener(this);
        b7 = new Button("6");
        b7.addActionListener(this);
        b8 = new Button("7");
        b8.addActionListener(this);
        b9 = new Button("8");
        b9.addActionListener(this);
        b10 = new Button("9");
        b10.addActionListener(this);
        b11 = new Button("+");
        b11.addActionListener(this);
        b12 = new Button("-");
        b12.addActionListener(this);
        b13 = new Button("*");
        b13.addActionListener(this);
        b14 = new Button("/");
```

```
b14.addActionListener(this);
b15 = newButton("%");
b15.addActionListener(this);
b16 = newButton("=");
b16.addActionListener(this);
b17 = new Button("C");
b17.addActionListener(this);
tf = newTextField(20);
f.add(tf);
g = newGridLayout(4,4,10,20);
p.setLayout(g);
p.add(b1);p.add(b2);p.add(b3);p.add(b4);p.add(b5);p.add(b6);p.add(b7);p.add(b8);p.add(b9);
p.add(b10);p.add(b11);p.add(b12);p.add(b13);p.add(b14);p.add(b15);p.add(b16);p.add(b17);
f.add(p);
f.setSize(300,300);
f.setVisible(true);
}
publicvoidactionPerformed(ActionEvent e)
{
    if(e.getSource()==b1)
    {
        s3 = tf.getText();
        s4 = "0";
        s5 = s3+s4;
        tf.setText(s5);
    }
    if(e.getSource()==b2)
    {
        s3 = tf.getText();
        s4 = "1";
        s5 = s3+s4;
        tf.setText(s5);
    }
    if(e.getSource()==b3)
    {
        s3 = tf.getText();
        s4 = "2";
        s5 = s3+s4;
        tf.setText(s5);
    }
    if(e.getSource()==b4)
    {
        s3 = tf.getText();
        s4 = "3";
        s5 = s3+s4;
        tf.setText(s5);
    }
    if(e.getSource()==b5)
    {
```

```
s3 = tf.getText();
s4 = "4";
s5 = s3+s4;
tf.setText(s5);
}
if(e.getSource()==b6)
{
s3 = tf.getText();
s4 = "5";
s5 = s3+s4;
tf.setText(s5);
}
if(e.getSource()==b7)
{
s3 = tf.getText();
s4 = "6";
s5 = s3+s4;
tf.setText(s5);
}
if(e.getSource()==b8)
{
s3 = tf.getText();
s4 = "7";
s5 = s3+s4;
tf.setText(s5);
}
if(e.getSource()==b9)
{
s3 = tf.getText();
s4 = "8";
s5 = s3+s4;
tf.setText(s5);
}
if(e.getSource()==b10)
{
s3 = tf.getText();
s4 = "9";
s5 = s3+s4;
tf.setText(s5);
}
if(e.getSource()==b11)
{
s1 = tf.getText();
tf.setText("");
c=1;
}
if(e.getSource()==b12)
{
```

```
s1 = tf.getText();
tf.setText("");
c=2;
}
if(e.getSource()==b13)
{
s1 = tf.getText();
tf.setText("");
c=3;

}
if(e.getSource()==b14)
{
s1 = tf.getText();
tf.setText("");
c=4;
}
if(e.getSource()==b15)
{
s1 = tf.getText();
tf.setText("");
c=5;
}
if(e.getSource()==b16)
{
s2 = tf.getText();
if(c==1)
{
n = Integer.parseInt(s1)+Integer.parseInt(s2);
tf.setText(String.valueOf(n));
}
else
if(c==2)
{
n = Integer.parseInt(s1)-Integer.parseInt(s2);
tf.setText(String.valueOf(n));
}
else
if(c==3)
{
n = Integer.parseInt(s1)*Integer.parseInt(s2);
tf.setText(String.valueOf(n));
}
if(c==4)
{
try
{
int p=Integer.parseInt(s2);
```

```
if(p!=0)
{
    n = Integer.parseInt(s1)/Integer.parseInt(s2);
    tf.setText(String.valueOf(n));
}
else
    tf.setText("infinite");
}
catch(Exception i){}
}
if(c==5)
{
    n = Integer.parseInt(s1)%Integer.parseInt(s2);
    tf.setText(String.valueOf(n));
}
}
if(e.getSource()==b17)
{
    tf.setText("");
}
}
public static void main(String[] abc)
{
    Calculator y = new Calculator();
}
}
```

Output:-

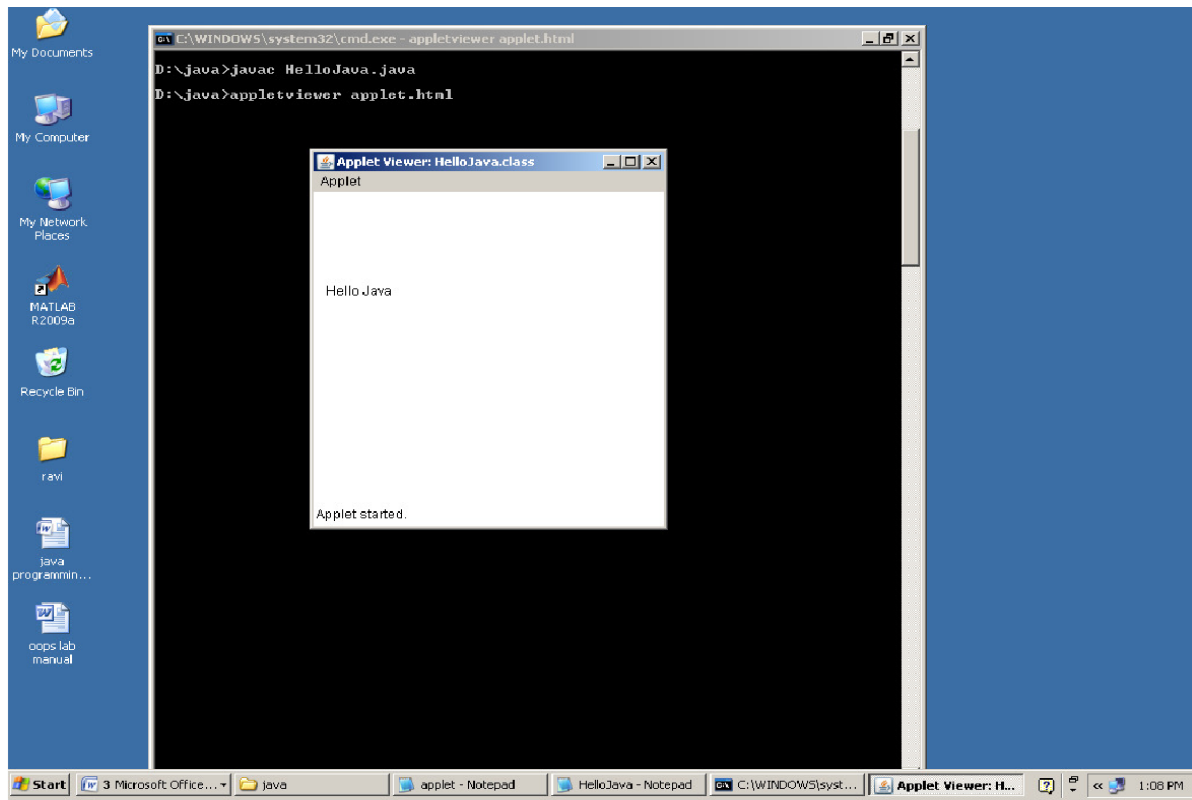


3) a) **Develop an applet that displays a simple message.**

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

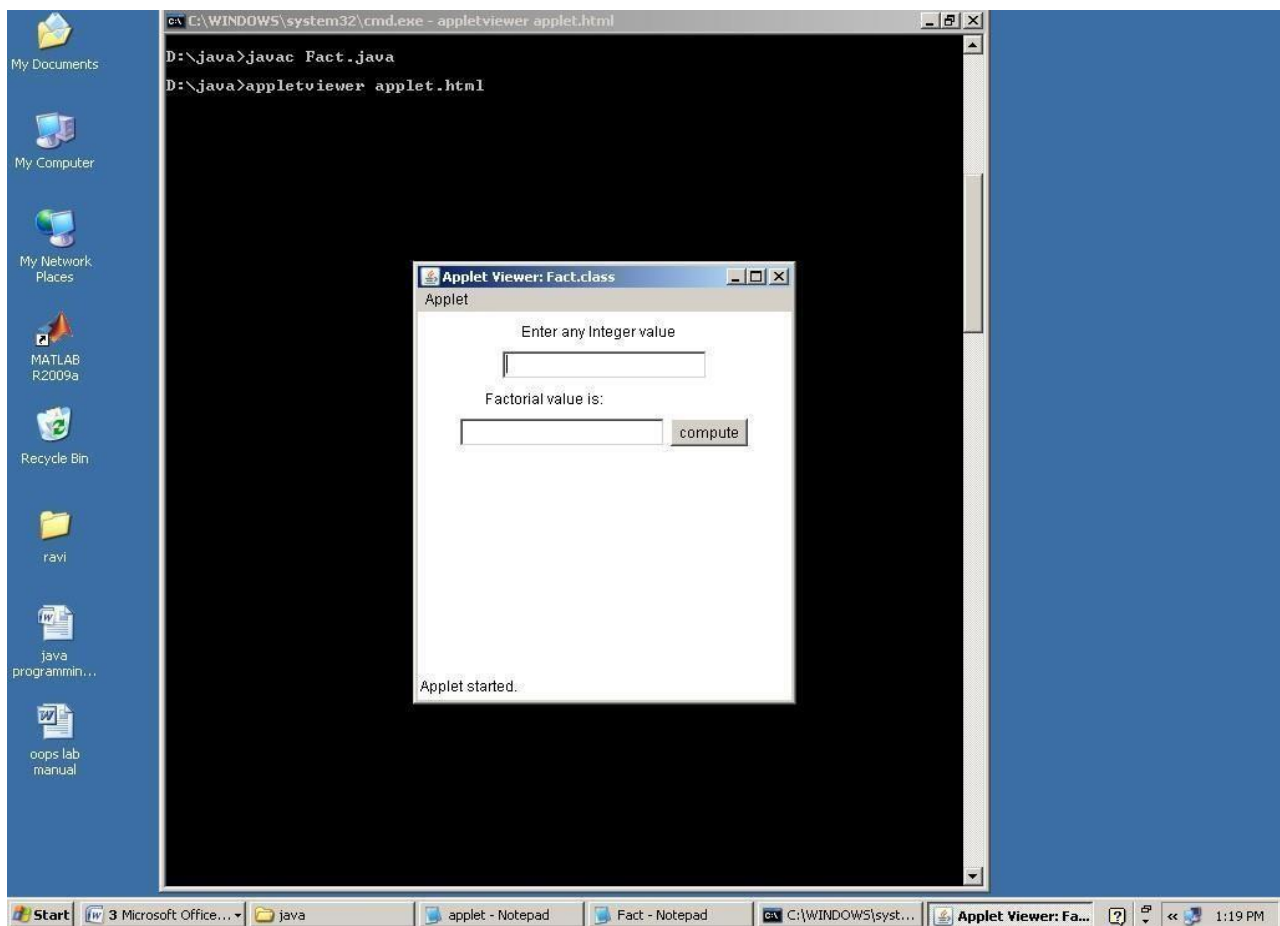
in another text field when the button "Compute" is clicked.

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 = "Arithmetic Exception "; repaint();
    }
    else if ((num1 < 0) || (num2 < 0))
    {
        msg = "NumberFormatException"; 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>

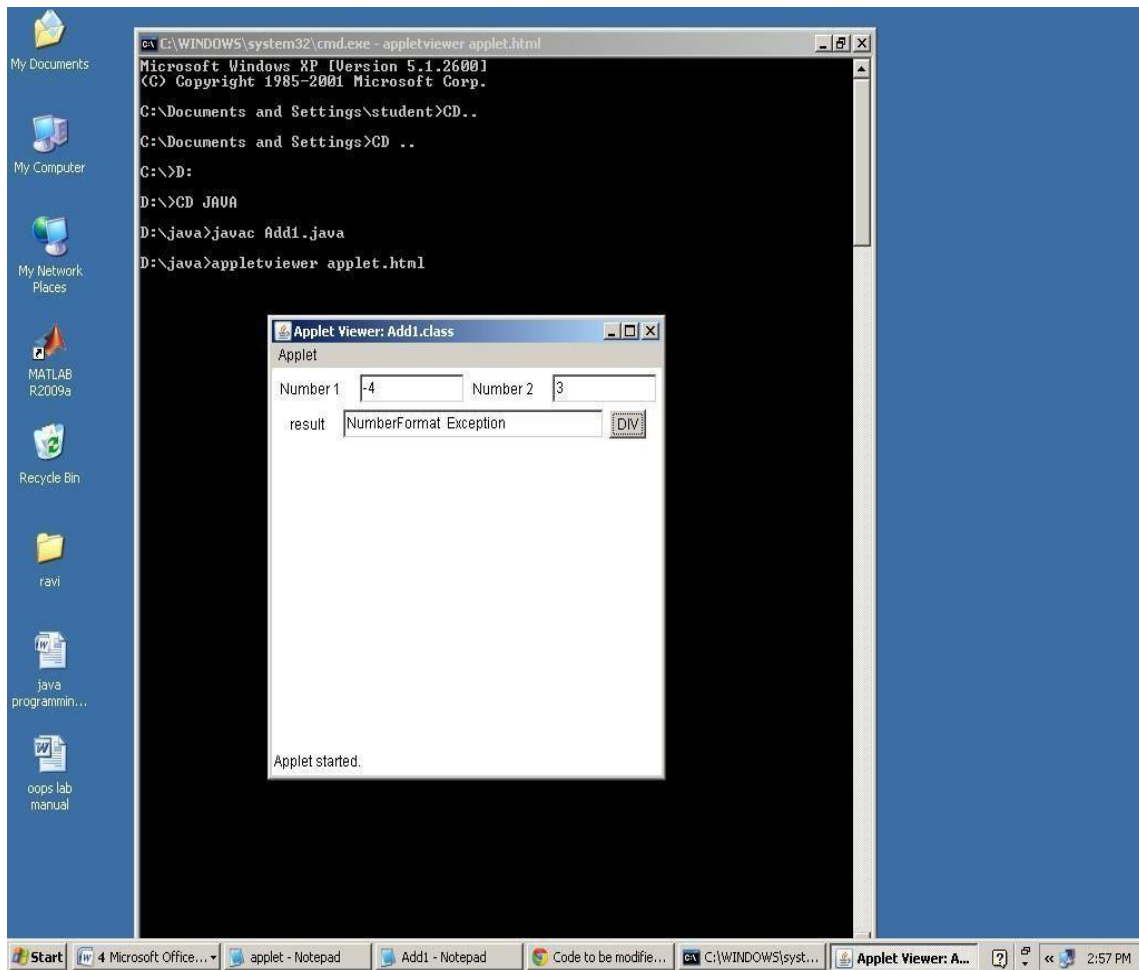
/*<applet code="Add1.class"width=350 height=300>

</applet>*/

</body>

</html>
```

Output:-



The screenshot displays a Windows XP desktop environment. A command prompt window is open, showing the following commands and output:

```
C:\WINDOWS\system32\cmd.exe - appletviewer applet.html
Microsoft Windows XP [Version 5.1.2600.1
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\student>CD ..
C:\Documents and Settings>CD ..
C:\>D:
D:\>CD J000
D:\java>javac Add1.java
D:\java>appletviewer applet.html
```

An "Applet Viewer: Add1.class" window is also open, showing a Java applet interface. The applet has two input fields: "Number 1" with the value "-4" and "Number 2" with the value "3". Below these fields is a "result" field containing the text "NumberFormatException" and a "DIV" button. The applet viewer also displays the message "Applet started." at the bottom.

The taskbar at the bottom shows several open applications: Microsoft Office Word, applet - Notepad, Add1 - Notepad, Code to be modify..., C:\WINDOWS\system..., and Applet Viewer: A... The system clock shows the time as 2:57 PM.

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:-

```
import java.util.Random;

class RandomNumberThread extends Thread {
    public void run() {
        Random random = new Random();
        for (int i = 0; i < 10; i++) {
            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 randomNumber) {
        number = randomNumber;
    }

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

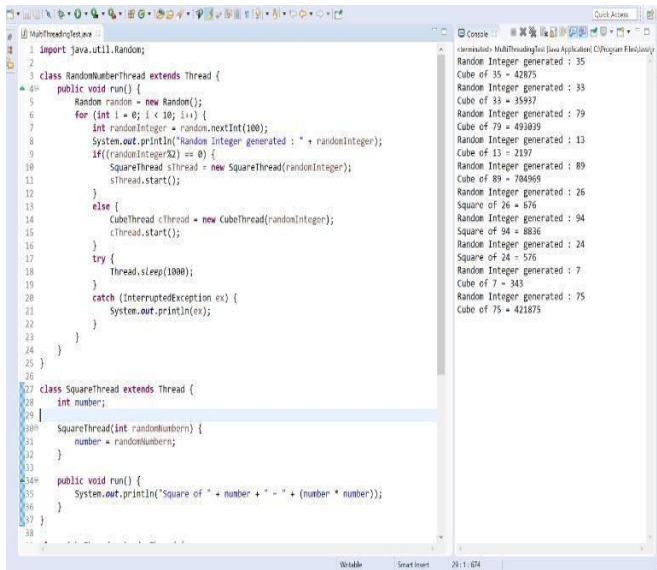
class CubeThread extends Thread {
    int number;

    CubeThread(int randomNumber) {
        number = randomNumber;
    }

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

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

Output:



```
1 import java.util.Random;
2
3 class RandomNumberThread extends Thread {
4     public void run() {
5         Random random = new Random();
6         for (int i = 0; i < 10; i++) {
7             int randomNumber = random.nextInt(100);
8             System.out.println("Random Integer generated : " + randomNumber);
9             if ((randomInteger % 2) == 0) {
10                SquareThread sThread = new SquareThread(randomInteger);
11                sThread.start();
12            }
13            else {
14                CubeThread cThread = new CubeThread(randomInteger);
15                cThread.start();
16            }
17            try {
18                Thread.sleep(1000);
19            }
20            catch (InterruptedException ex) {
21                System.out.println(ex);
22            }
23        }
24    }
25 }
26
27 class SquareThread extends Thread {
28     int number;
29
30     SquareThread(int randomNumber) {
31         number = randomNumber;
32     }
33
34     public void run() {
35         System.out.println("Square of " + number + " = " + (number * number));
36     }
37 }
38
```

```
Console
C:\Program Files\Java\jdk-1.8.0_101\bin\java.exe -classpath C:\Program Files\Java\jdk-1.8.0_101\bin\java.exe
Random Integer generated : 35
Cube of 35 = 42875
Random Integer generated : 33
Cube of 33 = 35937
Random Integer generated : 79
Cube of 79 = 493039
Random Integer generated : 13
Cube of 13 = 2197
Random Integer generated : 89
Cube of 89 = 704969
Random Integer generated : 26
Square of 26 = 676
Random Integer generated : 94
Square of 94 = 8836
Random Integer generated : 24
Square of 24 = 576
Random Integer generated : 7
Cube of 7 = 343
Random Integer generated : 75
Cube of 75 = 421875
```

6. Write a Java program for the following:

Create a doubly linked list of elements.

Delete a given element from the above list.

Display the contents of the list after deletion.

```
// Java program to delete a node from
// Doubly Linked List

// Class for Doubly Linked List
public class DLL {
    Node head; // head of list

    /* Doubly Linked list Node*/
    class Node {
        int data;
        Node prev;
        Node next;

        // Constructor to create a new node
        // next and prev is by default initialized
        // as null
        Node(int d) { data = d; }
    }

    // Adding a node at the front of the list
    public void push(int new_data)
    {
        // 1. allocate node
        // 2. put in the data
        Node new_Node = new Node(new_data);

        // 3. Make next of new node as head
        // and previous as NULL
        new_Node.next = head;
        new_Node.prev = null;

        // 4. change prev of head node to new node
        if (head != null)
            head.prev = new_Node;

        // 5. move the head to point to the new node
        head = new_Node;
    }

    // This function prints contents of linked list
```

```
// starting from the given node
public void printlist(Node node)
{
    Node last = null;

    while (node != null) {
        System.out.print(node.data + " ");
        last = node;
        node = node.next;
    }

    System.out.println();
}

// Function to delete a node in a Doubly Linked List.
// head_ref --> pointer to head node pointer.
// del --> data of node to be deleted.
void deleteNode(Node del)
{
    // Base case
    if (head == null || del == null) {
        return;
    }

    // If node to be deleted is head node
    if (head == del) {
        head = del.next;
    }

    // Change next only if node to be deleted
    // is NOT the last node
    if (del.next != null) {
        del.next.prev = del.prev;
    }

    // Change prev only if node to be deleted
    // is NOT the first node
    if (del.prev != null) {
        del.prev.next = del.next;
    }

    // Finally, free the memory occupied by del
    return;
}

// Driver Code
public static void main(String[] args)
```

```
{
// Start with the empty list
DLL dll = new DLL();

// Insert 2. So linked list becomes 2->NULL
dll.push(2);

// Insert 4. So linked list becomes 4->2->NULL
dll.push(4);

// Insert 8. So linked list becomes 8->4->2->NULL
dll.push(8);

// Insert 10. So linked list becomes 10->8->4->2->NULL
dll.push(10);

System.out.print("Created DLL is: ");
dll.printlist(dll.head);

// Deleting first node
dll.deleteNode(dll.head);

// List after deleting first node
// 8->4->2
System.out.print("\nList after deleting first node: ");
dll.printlist(dll.head);

// Deleting middle node from 8->4->2
dll.deleteNode(dll.head.next);

System.out.print("\nList after Deleting middle node: ");
dll.printlist(dll.head);
}
}
```

Output:

Original Linked list 10 8 4 2

Modified Linked list 8

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 selected color. Initially, there is no message shown.

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;

/*
 * <applet code = "TrafficLightsExample" width = 1000 height = 500>
 * </applet>
 * */

public class TrafficLightsExample extends Applet implements ItemListener{

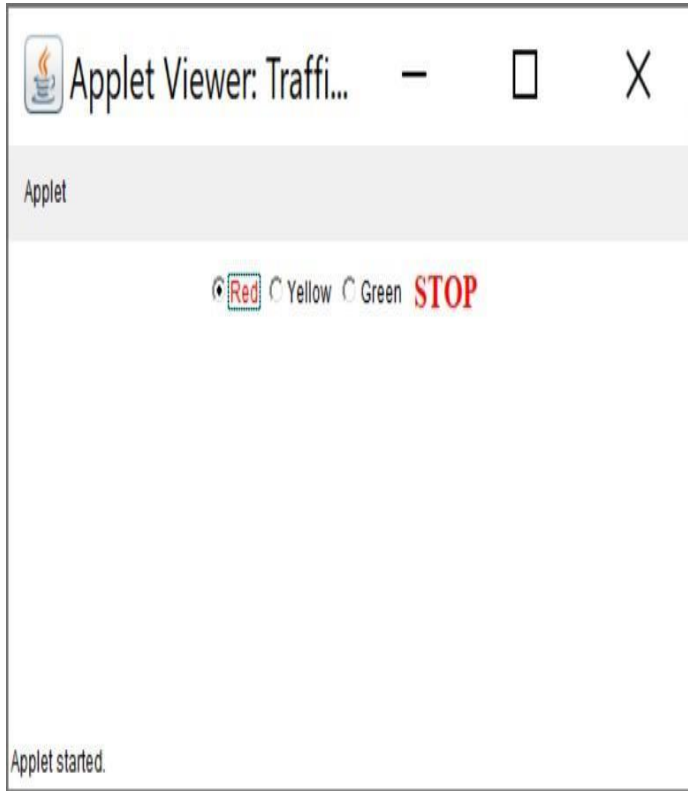
    CheckboxGroup grp = new CheckboxGroup();
    Checkbox redLight, yellowLight, greenLight;
    Label msg;
    public void init(){
        redLight = new Checkbox("Red", grp, false);
        yellowLight = new Checkbox("Yellow", grp, false);
        greenLight = new Checkbox("Green", grp, false);
        msg = new Label("");

        redLight.addItemListener(this);
        yellowLight.addItemListener(this);
        greenLight.addItemListener(this);

        add(redLight);
        add(yellowLight);
    }
}
```

```
        add(greenLight);
        add(msg);
        msg.setFont(new Font("Serif", Font.BOLD, 20));
    }
    public void itemStateChanged(ItemEvent ie) {
        redLight.setForeground(Color.BLACK);
        yellowLight.setForeground(Color.BLACK);
        greenLight.setForeground(Color.BLACK);

        if(redLight.getState() == true) {
            redLight.setForeground(Color.RED);
            msg.setForeground(Color.RED);
            msg.setText("STOP");
        }
        else if(yellowLight.getState() == true) {
            yellowLight.setForeground(Color.YELLOW);
            msg.setForeground(Color.YELLOW);
            msg.setText("READY");
        }
        else{
            greenLight.setForeground(Color.GREEN);
            msg.setForeground(Color.GREEN);
            msg.setText("GO");
        }
    }
}
```



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

```
import java.util.*;

abstract class Shape {
    int length, breadth, radius;

    Scanner input = new Scanner(System.in);

    abstract void printArea();
}

class Rectangle extends Shape {
    void printArea() {
        System.out.println("*** Finding the Area of Rectangle ***");
        System.out.print("Enter length and breadth: ");
        length = input.nextInt();
        breadth = input.nextInt();
        System.out.println("The area of Rectangle is: " + length * breadth);
    }
}

class Triangle extends Shape {
    void printArea() {
        System.out.println("\n*** Finding the Area of Triangle ***");
        System.out.print("Enter Base And Height: ");
        length = input.nextInt();
        breadth = input.nextInt();
        System.out.println("The area of Triangle is: " + (length * breadth) / 2);
    }
}
```

```
class Cricle extends Shape {
    void printArea() {
        System.out.println("\n*** Finding the Area of Cricle ***");

System.out.print("Enter Radius: ");
        radius = input.nextInt();
        System.out.println("The area of Cricle is: " + 3.14f * radius * radius);
    }
}

public class AbstractClassExample {
    public static void main(String[] args) {
        Rectangle rec = new Rectangle();
        rec.printArea();

        Triangle tri = new Triangle();
        tri.printArea();

        Cricle cri = new Cricle();
        cri.printArea();
    }
}
```

Output:



```

AbstractClassExample.java
19     }
20 }
21
22 class Triangle extends Shape {
23     void printArea() {
24         System.out.println("\n*** Finding the Area:");
25         System.out.print("Enter Base And Height: ");
26         length = input.nextInt();
27         breadth = input.nextInt();
28         System.out.println("The area of Triangle is: ");
29     }
30 }
31
32 class Cricle extends Shape {
33     void printArea() {
34         System.out.println("\n*** Finding the Area:");
35         System.out.print("Enter Radius: ");
36         radius = input.nextInt();
37         System.out.println("The area of Cricle is: ");
38     }
39 }
40
41 public class AbstractClassExample {
42     public static void main(String[] args) {
43         Rectangle rec = new Rectangle();
44         rec.printArea();
    }
}

Console
C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe [2]
*** Finding the Area of Rectangle ***
Enter length and breadth: 2 3
The area of Rectangle is: 6

*** Finding the Area of Triangle ***
Enter Base And Height: 4 5
The area of Triangle is: 10

*** Finding the Area of Cricle ***
Enter Radius: 5
The area of Cricle is: 78.5
    
```

9. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in Grid Layout.

Source code:

```

import java.io.*;
import java.util.*;
import java.awt.*;
import javax.swing.*;

class A extends JFrame {
    public A() {
        setSize(400, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        GridLayout g = new GridLayout(0, 3);
        setLayout(g);
        try {
            FileInputStream fin = new
            FileInputStream("C:\\Users\\User\\eclipse-workspace\\LabManual\\src\\HashTab.txt");
            Scanner sc = new Scanner(fin).useDelimiter(",");
            String[] arrayList;
            String a;
            while (sc.hasNextLine()) {
                a = sc.nextLine(); arrayList =
                a.split(","); for (String i : arrayList) {
                    add(new JLabel(i));
                }
            }
        } catch (Exception ex) {
        }
        setDefaultCloseOperation(true);
        pack();
        setVisible(true);
    }
}
    
```



```
public class TableTest {
```

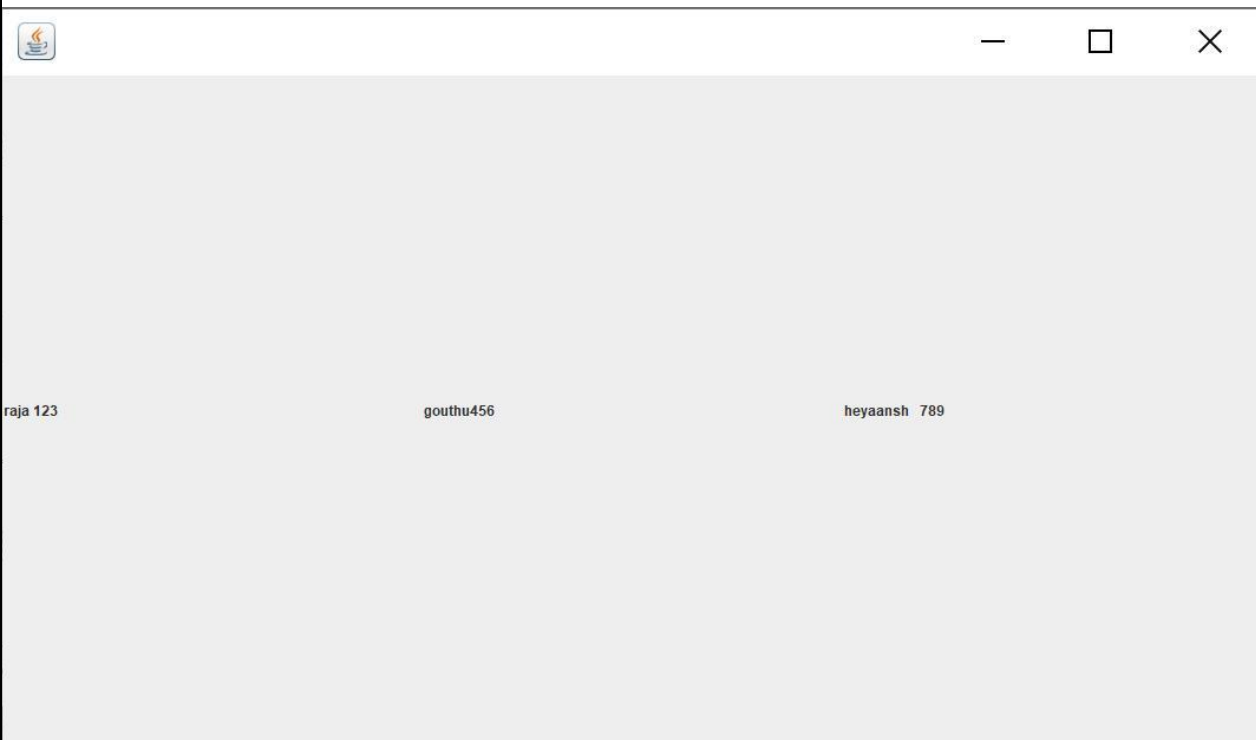
```
    public static void main(String[] args) {
```

```
        A a = new A();
```

```
    }
```

```
}
```

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).

Source code:

```
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)

Source code:

```
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.readFile("HashTab.txt");
        System.out.println("File data into Hashtable:\n" + hashData); prog11.printTheData(hashData,
"raja"); 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> readFile(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;
}
}
```

Output:

```
C:\WINDOWS\system32\cmd.exe
Data not found in HashTable
D:\java>javac HashTab.java
D:\java>java HashTab
File data into HashTable:
CodeF=567, abc=345, vbit=123
Input given:vbit
Data found in HashTable:123
Input given:123
Data found in HashTable:vbit
D:\java>_
```

12. Write a Java program that correctly implements the producer – consumer problem using the concept of interthread communication.

Source Code:

```
class ItemQueue {
    int item;
    boolean valueSet = false;

    synchronized int getItem()
    {
        while (!valueSet)
            try {
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        System.out.println("Consummed:" + item);
        valueSet = false;
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
        }
        notify();
        return item;
    }

    synchronized void putItem(int item) {
        while (valueSet)
            try {
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        this.item = item;
        valueSet = true;
        System.out.println("Produced: " + item);
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
        }
        notify();
    }
}

class Producer implements Runnable{
    ItemQueue itemQueue;
    Producer(ItemQueue itemQueue){
        this.itemQueue = itemQueue;
        new Thread(this, "Producer").start();
    }
}
```



```
public void run() {
    int i = 0;
    while(true) {
        itemQueue.putItem(i++);
    }
}

class Consumer implements Runnable{

    ItemQueue itemQueue;
    Consumer(ItemQueue itemQueue){
        this.itemQueue = itemQueue;
        new Thread(this, "Consumer").start();
    }

    public void run() {
        while(true) {
            itemQueue.getItem();
        }
    }
}

class ProducerConsumer{
    public static void main(String args[]) { ItemQueue itemQueue
        = new ItemQueue(); new Producer(itemQueue);
        new Consumer(itemQueue);
    }
}
```

Output:

The screenshot shows the Eclipse IDE with the following code in the editor:

```
1 class ItemQueue {
2     int item;
3     boolean valueSet = false;
4
5     synchronized int getItem()
6
7     {
8         while (!valueSet)
9             try {
10                wait();
11            } catch (InterruptedException e) {
12                System.out.println("InterruptedException caught");
13            }
14        System.out.println("Consumed:" + item);
15        valueSet = false;
16        try {
17            Thread.sleep(1000);
18        } catch (InterruptedException e) {
19            System.out.println("InterruptedException caught");
20        }
21        notify();
22        return item;
23    }
24
25    synchronized void putItem(int item) {
26        while (valueSet)
27            try {
28                wait();
29            } catch (InterruptedException e) {
30                System.out.println("InterruptedException caught");
31            }
32        this.item = item;
33        valueSet = true;
34        System.out.println("Produced: " + item);
35        try {
36            Thread.sleep(1000);
37        } catch (InterruptedException e) {
38            System.out.println("InterruptedException caught");
39        }
40        notify();
41    }
42 }
43 }
```

The console output shows the following sequence of events:

```
<terminated> ProducerConsumer (Java Application) C:\Program
Produced: 0
Consummed:0
Produced: 1
Consummed:1
Produced: 2
Consummed:2
Produced: 3
Consummed:3
Produced: 4
Consummed:4
Produced: 5
Consummed:5
Produced: 6
Consummed:6
Produced: 7
Consummed:7
Produced: 8
Consummed:8
Produced: 9
Consummed:9
Produced: 10
Consummed:10
Produced: 11
Consummed:11
Produced: 12
Consummed:12
Produced: 13
Consummed:13
Produced: 14
Consummed:14
Produced: 15
Consummed:15
Produced: 16
Consummed:16
Produced: 17
Consummed:17
Produced: 18
Consummed:18
Produced: 19
Consummed:19
Produced: 20
```

13. Write a Java program to list all the files in a directory including the files present in all its subdirectories.

Source Code:

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

public class ListingFiles {

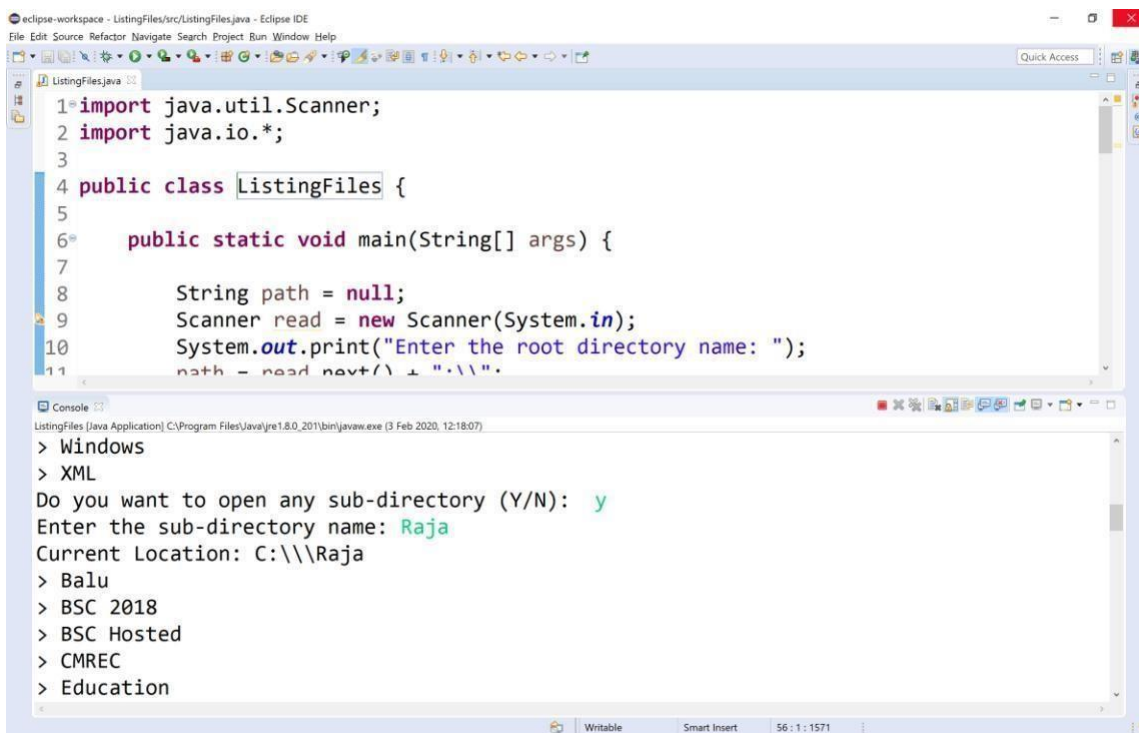
    public static void main(String[] args) {

        String path = null;
        Scanner read = new Scanner(System.in);
        System.out.print("Enter the root directory name: "); path =
        read.next() + "\\"; File f_ref = new File(path);
        if (!f_ref.exists()) {
            printLine();
            System.out.println("Root directory does not exists!"); printLine();
        }
        else {
            String ch = "y";
            while (ch.equalsIgnoreCase("y")) {
                printFiles(path);
                System.out.print("Do you want to open any sub-directory(Y/N): ");
                ch = read.next().toLowerCase();
                if (ch.equalsIgnoreCase("y")) {
                    System.out.print("Enter the sub-directory name: "); path = path +
                    "\\\" + read.next(); File f_ref_2 = new File(path);
                    if (!f_ref_2.exists()) {
                        printLine();
                        System.out.println("The sub-directory does not exists!");
                        printLine();
                        int lastIndex = path.lastIndexOf("\\");
                        path = path.substring(0, lastIndex);
                    }
                }
            }
        }
        System.out.println("***** Program Closed *****");
    }

    public static void printFiles(String path) {
        System.out.println("Current Location: " + path);
        File f_ref = new File(path);
        File[] filesList = f_ref.listFiles();
        for (File file : filesList) {
            if (file.isFile())
                System.out.println("- " + file.getName());
            else
        }
    }
}
```

```
        System.out.println("> " + file.getName());
    }
}
public static void printLine() {
    System.out.println("-----");
}
}
```

Output:



The screenshot shows the Eclipse IDE with a Java file named ListingFiles.java. The code in the editor is as follows:

```
1 import java.util.Scanner;
2 import java.io.*;
3
4 public class ListingFiles {
5
6     public static void main(String[] args) {
7
8         String path = null;
9         Scanner read = new Scanner(System.in);
10        System.out.print("Enter the root directory name: ");
11        path = read.next() + "\\.";
12    }
13 }
```

The console output shows the execution of the program:

```
> Windows
> XML
Do you want to open any sub-directory (Y/N): y
Enter the sub-directory name: Raja
Current Location: C:\\\\Raja
> Balu
> BSC 2018
> BSC Hosted
> CMREC
> Education
```

14. Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending Order.

Source Code:

public class QuickSortOnStrings {

String names[];

int length;

public static void main(String[] args) { QuickSortOnStrings obj = **new**
QuickSortOnStrings();

String stringsList[] = {"raja", "gouthu", "rani", "gouthami", "honey", "heyaansh", "hello"};
obj.sort(stringsList);

for (String i : stringsList) {
 System.**out**.print(i);
 System.**out**.print(" ");
}

}

void sort(String array[]) {

if (array == **null** || array.length == 0) {
 return;

}

this.names = array;

this.length = array.length;

quickSort(0, length - 1);

}

void quickSort(**int** lowerIndex, **int** higherIndex) { **int** i =
lowerIndex;

int j = higherIndex;

String pivot = **this.names**[lowerIndex + (higherIndex - lowerIndex) / 2];

while (i <= j) {

while (**this.names**[i].compareToIgnoreCase(pivot) < 0) { i++;
 }

while (**this.names**[j].compareToIgnoreCase(pivot) > 0) { j--;
 }

if (i <= j) {
 exchangeNames(i, j);
 i++;
 j--;
 }

}

if (lowerIndex < j) {

 quickSort(lowerIndex, j);

}

if (i < higherIndex) {

 quickSort(i, higherIndex);

}

}

void exchangeNames(**int** i, **int** j)

{

String temp = **this.names**[i];

this.names[i] = **this.names**[j];

this.names[j] = temp;

}

}

Output:

```

1 public class QuickSortOnStrings {
2
3
4     String names[];
5     int length;
6
7     public static void main(String[] args) {
8         QuickSortOnStrings obj = new QuickSortOnStrings();
9         String stringsList[] = {"raja", "gouthu", "rani", "gouthami", "honey", "heyaansh", "hello"};
10        obj.sort(stringsList);
11
12        for (String i : stringsList) {
13            System.out.print(i);
14            System.out.print(" ");
15        }
16    }
17
18    void sort(String array[]) {

```

Console Output:
gouthami gouthu hello heyaansh honey raja rani |

15. Write a Java program that implements Bubble sort algorithm for sorting in descending order and also shows the number of interchanges occurred for the given set of integers.

Source Code:

```

import java.util.Scanner;
public class BubbleSort {

    public static void main(String[] args) { Scanner read = new
        Scanner(System.in); int size, count = 0;
        //Reading size of the list
        System.out.print("Enter the list size: ");
        size = read.nextInt();

        //Creating list with elements
        int list[] = new int[size];
        System.out.println("Enter any " + size + " integer numbers: "); for(int i = 0; i <
        size; i++)
            list[i] = read.nextInt();

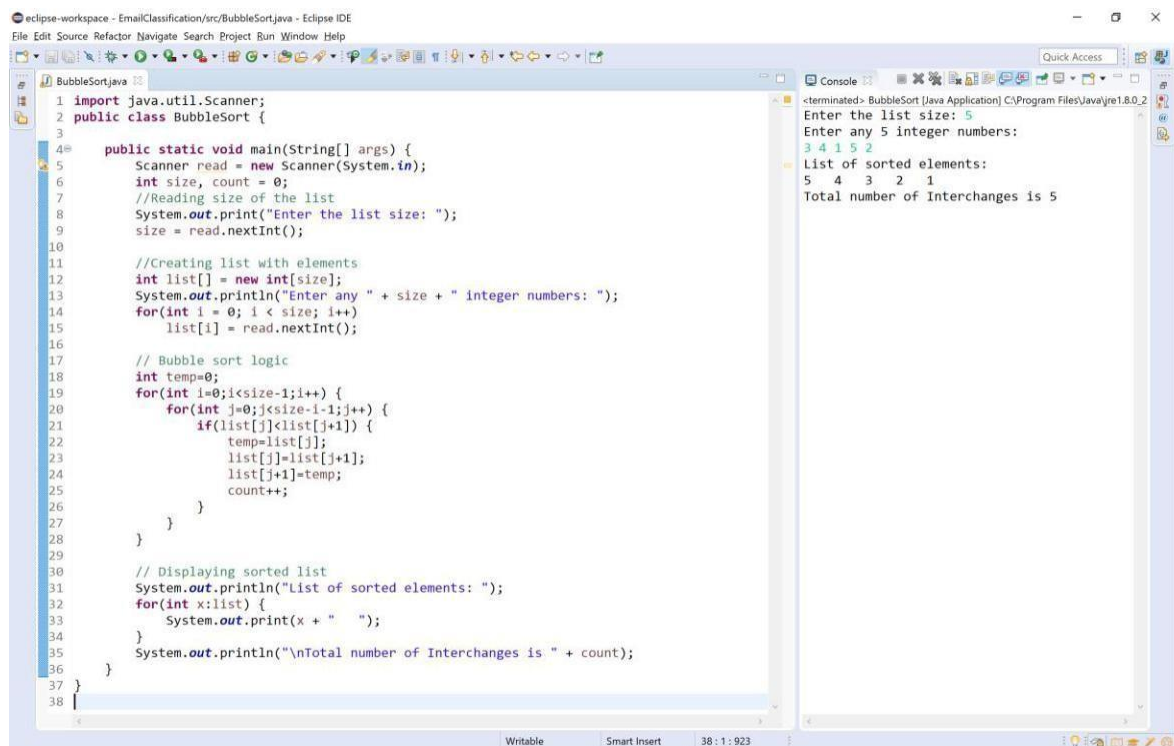
        // Bubble sort logic
        int temp=0;
        for(int i=0;i<size-1;i++) { for(int j=0;j<size-i-
            1;j++) {
                if(list[j]<list[j+1]) {
                    temp=list[j];
                    list[j]=list[j+1];
                    list[j+1]=temp;
                    count++;
                }
            }
        }
        System.out.println("Number of interchanges: " + count);
    }
}

```

```
        temp=list[j];
        list[j]=list[j+1];
        list[j+1]=temp;
        count++;
    }
}

// Displaying sorted list
System.out.println("List of sorted elements: ");
for(int x:list) {
    System.out.print(x + " ");
}
System.out.println("\nTotal number of Interchanges is " + count);
}
}
```

Output:



```
eclipse-workspace - EmailClassification/arc/BubbleSort.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

BubbleSort.java
1 import java.util.Scanner;
2 public class BubbleSort {
3
4     public static void main(String[] args) {
5         Scanner read = new Scanner(System.in);
6         int size, count = 0;
7         //Reading size of the list
8         System.out.print("Enter the list size: ");
9         size = read.nextInt();
10
11         //Creating list with elements
12         int list[] = new int[size];
13         System.out.println("Enter any " + size + " integer numbers: ");
14         for(int i = 0; i < size; i++)
15             list[i] = read.nextInt();
16
17         // Bubble sort logic
18         int temp=0;
19         for(int i=0;i<size-1;i++) {
20             for(int j=0;j<size-1-i;j++) {
21                 if(list[j]>list[j+1]) {
22                     temp=list[j];
23                     list[j]=list[j+1];
24                     list[j+1]=temp;
25                     count++;
26                 }
27             }
28         }
29
30         // Displaying sorted list
31         System.out.println("List of sorted elements: ");
32         for(int x:list) {
33             System.out.print(x + " ");
34         }
35         System.out.println("\nTotal number of Interchanges is " + count);
36     }
37 }
38

Console
terminated> BubbleSort [Java Application] C:\Program Files\Java\jre1.8.0_2
Enter the list size: 5
Enter any 5 integer numbers:
3 4 1 5 2
List of sorted elements:
5 4 3 2 1
Total number of Interchanges is 5

Writable Smart Insert 38 : 1: 923
```

Additional Programs

1. Write a java program that connects to a database using JDBC

Aim: To Write a java program that connects to a database using JDBC

Source Code:

```
import java.sql.*;
class MysqlCon{
public static void main(String args[]){
try{
Class.forName("com.mysql.jdbc.Driver");
Connection con=DriverManager.getConnection(
"jdbc:mysql://localhost:3306/sonoo","root","root");
//here sonoo is database name, root is username and password
Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery("select * from emp");
while(rs.next())
System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3));
con.close();
}catch(Exception e){ System.out.println(e);}
}
}
```

2. Write a java program to connect to a database using JDBC and insert values into it

Aim: To Write a java program to connect to a database using JDBC and insert values into it

Source Code:

```
import java.sql.*;
```

```
// Main/App class of above Connection class
```

```
public class GFG {
```

```
// MAin driver method
```

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

```
{
```

```
    // Step 2: Showing above Connection class i.e
```

```
    // loading and registering drivers
```

```
    // Initially assigning NULL parameters
```

```
    // to object of Connection class
```

```
    Connection con = null;
```

```
    PreparedStatement ps = null;
```

```
    // Step 3: Establish the connection
```

```
    con.connectDB();
```

```
// Try block to check if exception/s occurs
try {

    // Step 4: Create a statement
    String sql = "insert into cuslogin values('geeksforgeeks','gfg','geeks@email.com','flat 1','1239087474',10)";

    // Step 5: Execute the query
    ps = con.prepareStatement(sql);

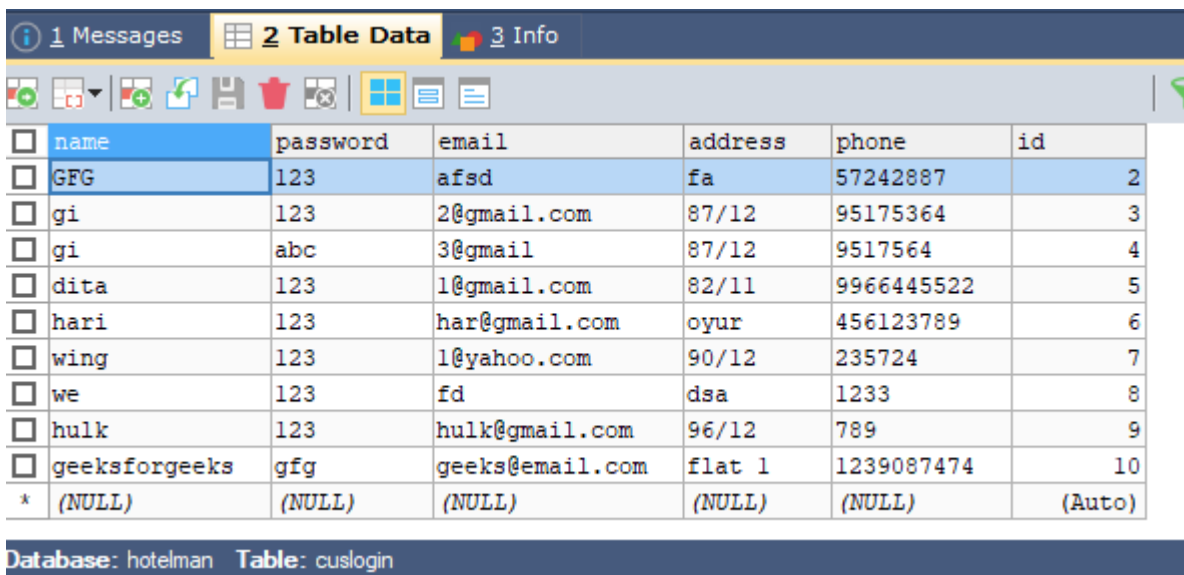
    // Step 6: Process the results
    ps.execute();
}

// Optional but recommended
// Step 7: Close the connection

// Catch block to handle the exception/s
catch (Exception e) {

    // Print the exception
    System.out.println(e);
}
}
```

Output:



name	password	email	address	phone	id
GFG	123	afsd	fa	57242887	2
gi	123	2@gmail.com	87/12	95175364	3
gi	abc	3@gmail	87/12	9517564	4
dita	123	1@gmail.com	82/11	9966445522	5
hari	123	har@gmail.com	oyur	456123789	6
wing	123	1@yahoo.com	90/12	235724	7
we	123	fd	dsa	1233	8
hulk	123	hulk@gmail.com	96/12	789	9
geeksforgeeks	gfg	geeks@email.com	flat 1	1239087474	10
* (NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(Auto)

Database: hotelman Table: cuslogin

3. Write a java program to connect to a database using JDBC and delete values from it

Aim: To Write a java program to connect to a database using JDBC and delete values from it

Source Code:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
```

```
public class JDBCExample {
    static final String DB_URL = "jdbc:mysql://localhost/TUTORIALSPOINT";
    static final String USER = "guest";
    static final String PASS = "guest123";
    static final String QUERY = "SELECT id, first, last, age FROM Registration";

    public static void main(String[] args) {
        // Open a connection
        try(Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
            Statement stmt = conn.createStatement();
        ) {
            String sql = "DELETE FROM Registration " +
                "WHERE id = 101";
            stmt.executeUpdate(sql);
            ResultSet rs = stmt.executeQuery(QUERY);
            while(rs.next()){
                //Display values
                System.out.print("ID: " + rs.getInt("id"));
                System.out.print(", Age: " + rs.getInt("age"));
                System.out.print(", First: " + rs.getString("first"));
                System.out.println(", Last: " + rs.getString("last"));
            }
            rs.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Output:

```
ID: 100, Age: 30, First: Zara, Last: Ali
ID: 102, Age: 30, First: Zaid, Last: Khan
ID: 103, Age: 28, First: Sumit, Last: Mittal
```

4. Write a java program for handling Mouse events

Aim: To Write a java program for handling Mouse events

Source Code:

```
import java.awt.*;

import java.awt.event.*;

import java.applet.*;

/*

<applet code="MouseEvents" width=300 height=100>

</applet>
*/
```

```
public class MouseEvents extends Applet

implements MouseListener, MouseMotionListener {

String msg = "";

int mouseX = 0, mouseY = 0; // coordinates of mouse

public void init() {

addMouseListener(this);

addMouseMotionListener(this);

}

// Handle mouse clicked.

public void mouseClicked(MouseEvent me) {

// save coordinates

mouseX = 0;

mouseY = 10;

msg = "Mouse clicked.";

repaint();

}

// Handle mouse entered.

public void mouseEntered(MouseEvent me) {

// save coordinates

mouseX = 0;

mouseY = 10;

msg = "Mouse entered.";

repaint();

}

// Handle mouse exited.

public void mouseExited(MouseEvent me) {

// save coordinates

mouseX = 0;
```

```
mouseY = 10;

msg = "Mouse exited.";

repaint();

}

// Handle button pressed.

public void mousePressed(MouseEvent me) {

// save coordinates

mouseX = me.getX();

mouseY = me.getY();

msg = "Down";

repaint();

}

// Handle button released.

public void mouseReleased(MouseEvent me) {

// save coordinates

mouseX = me.getX();

mouseY = me.getY();

msg = "Up";

repaint();

}

// Handle mouse dragged.

public void mouseDragged(MouseEvent me) {

// save coordinates

mouseX = me.getX();

mouseY = me.getY();

msg = "*";

showStatus("Dragging mouse at " + mouseX + ", " + mouseY);
```

```
}  
  
// Handle mouse moved.  
  
public void mouseMoved(MouseEvent me) {  
  
// show status  
  
showStatus("Moving mouse at " + me.getX() + ", " + me.getY());  
  
}  
  
// Display msg in applet window at current X,Y location.  
  
public void paint(Graphics g) {  
  
g.drawString(msg, mouseX, mouseY);  
  
}  
  
}
```

Output:

