

MALLA REDDY INSTITUTE OF ENGINEERING & TECHNOLOGY

(Sponsored by Malla Reddy Educational Society) Affiliated to JNTU, Accredited by NBA, Hyderabad)

Maisammaguda, Dhulapally (post via Hakimpet), Sec'Bad-500 014.

Phone: 040-65969674, Cell: 9348161223



MALLA REDDY INSTITUTE OF ENGINEERING & TECHNOLOGY

LABORATORY MANUAL

JAVA PROGRAMMING LAB B. Tech II YEAR-II SEM

Department of

INFORMATION TECHNOLOGY

INDEX

S.NO	NAME OF 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.	1-2
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.	3-9
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.	10-13
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.	14-16
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.	17-19
6	Write a Java program for the following: i) Create a doubly linked list of elements. ii) Delete a given element from the above list. iii) Display the contents of the list after deletion.	20-26
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.	27-29
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.	30-31

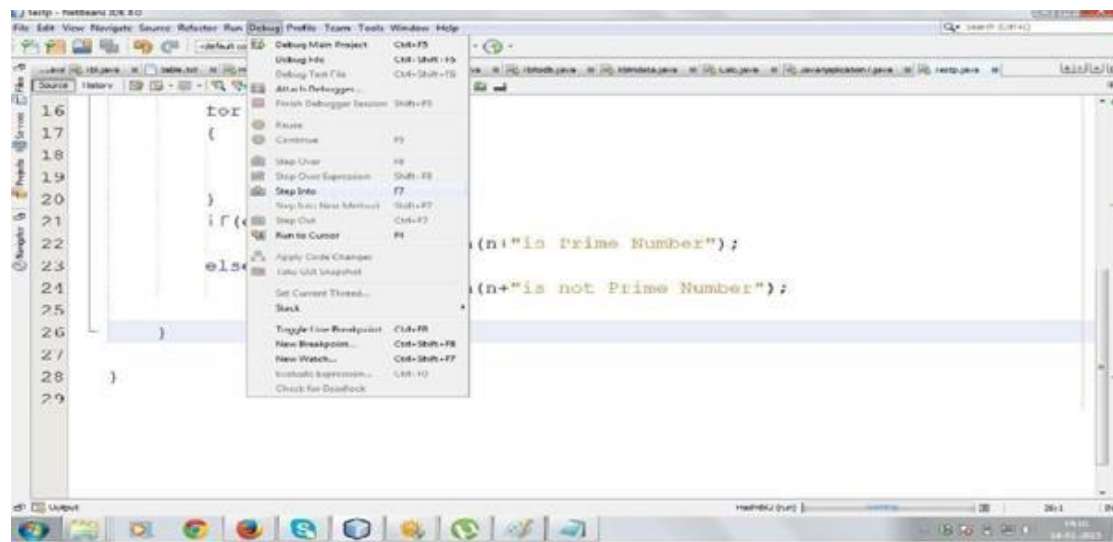
9	Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in Grid Layout.	32-33
10	Write a Java program 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).	34-35
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).	36-38
12	Write a Java program that correctly implements the producer – consumer problem using the concept of interthread communication.	39-41
13	Write a Java program to list all the files in a directory including the files present in all its subdirectories.	42-43
14	Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order	44-46
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.	47-48

EXPERIMENT NO: 01**Name of the Program:**

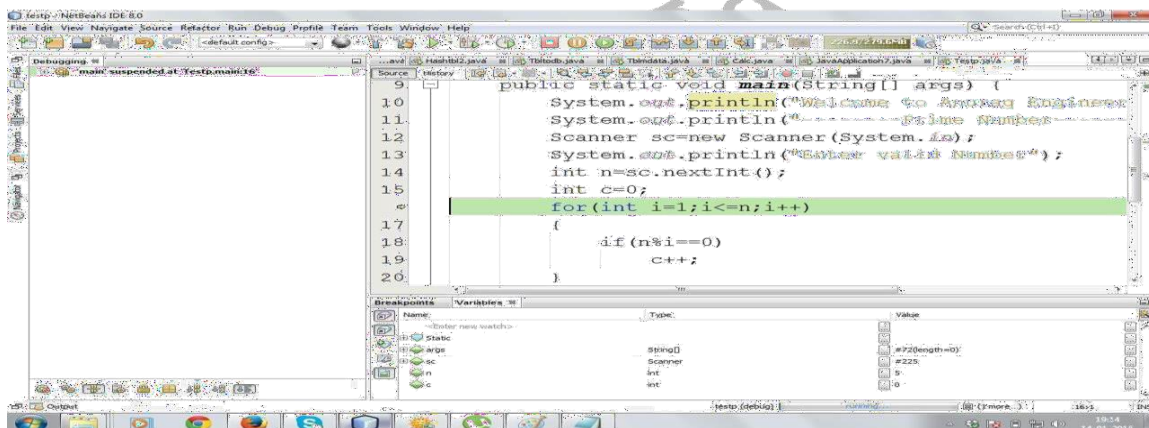
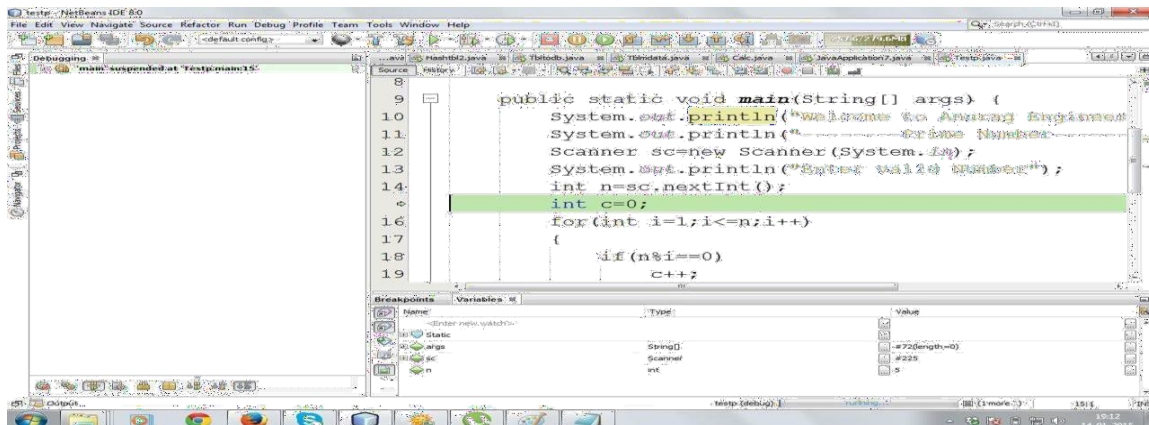
Use Eclipse or Net bean platform and acquaint with the various menus. Create a test project, add a test class and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with small program of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

Source Code:

```
import java.util.*;
public class Testp
{
public static void main(String[] args)
{
System.out.println ("\n Welcome to Malla Reddy Institute of Engineering and Technology
Students\n");
System.out.println("\n-----Prime Number----- \n");
Scanner sc = new Scanner(System.in);
System.out.println("\n Enter valid Number\n");
int n = sc.nextInt();
int c = 0;
for (int i = 1; i <= n; i++)
{
if (n % i == 0)
{
c++;
}
}
if (c == 2)
{
System.out.println(n + "is Prime Number \n");
}
else
{
System.out.println(n + "is not Prime Number\n");
}
}
}
```

Output:

Press F7 or select Step Into to start Debug



EXPERIMENT NO: 02**Name of the Program:**

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

Source Code:

```
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;
import java.awt.event.*;
class A extends JFrame implements ActionListener
{
    public JButton b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12, b13, b14, b15, b16;
    public JTextField tf1;
    public JPanel p;
    public String v = "";
    public String v1 ="0";
    public String op = "";
    public A()
    {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);    setSize(400, 400);
        p = new JPanel(new FlowLayout());
        tf1 = new JTextField(10);
        p.add(tf1);
        add(p);
        setLayout(new GridLayout(0, 3));
        b1 = new JButton("1");
        b1.addActionListener(this);
        add(b1);
        b2 = new JButton("2");
        b2.addActionListener(this);
        add(b2);
        b3 = new JButton("3");
        b3.addActionListener(this);
        add(b3);
        b4 = new JButton("4");
        b4.addActionListener(this);
        add(b4);
        b5 = new JButton("5");
        b5.addActionListener(this);
        add(b5);
        b6 = new JButton("6");
        b6.addActionListener(this);
        add(b6);
        b7 = new JButton("7");
```

```
b7.addActionListener(this);
add(b7);
b8 = new JButton("8");
b8.addActionListener(this);
add(b8);
b9 = new JButton("9");
b9.addActionListener(this);
add(b9);
b10 = new JButton("0");
b10.addActionListener(this);
add(b10);
b11 = new JButton("+");
b11.addActionListener(this);
add(b11);
b12 = new JButton("-");
b12.addActionListener(this);
add(b12);
b13 = new JButton("*");
b13.addActionListener(this);
add(b13);
b14 = new JButton("/");
b14.addActionListener(this);
add(b14);
b16 = new JButton("%");
b16.addActionListener(this);
add(b16);
b15 = new JButton("=");
b15.addActionListener(this);
add(b15);
setVisible(true);
}
public void actionPerformed(ActionEvent ae)
{
String b = ae.getActionCommand();
    switch (b)
    {
        case "1":
        {
            v = v + "1";
            tf1.setText(v);
        }

        break;

        case "2":
```

```
{
v = v + "2";
tf1.setText(v);
}
break;
case "3":
{
v = v + "3";
tf1.setText(v);
}
break;
case "4":
{
v = v + "4";
tf1.setText(v);
}
break;
case "5":
{
v = v + "5";
tf1.setText(v);
}
break;
case "6":
{
v = v + "6";
tf1.setText(v);
}
break;
case "7":
{
v = v + "7";
tf1.setText(v);
}
break;
case "8":
{
v = v + "8";
tf1.setText(v);
}
break;
case "9":
{
v = v + "9";
tf1.setText(v);
}
```

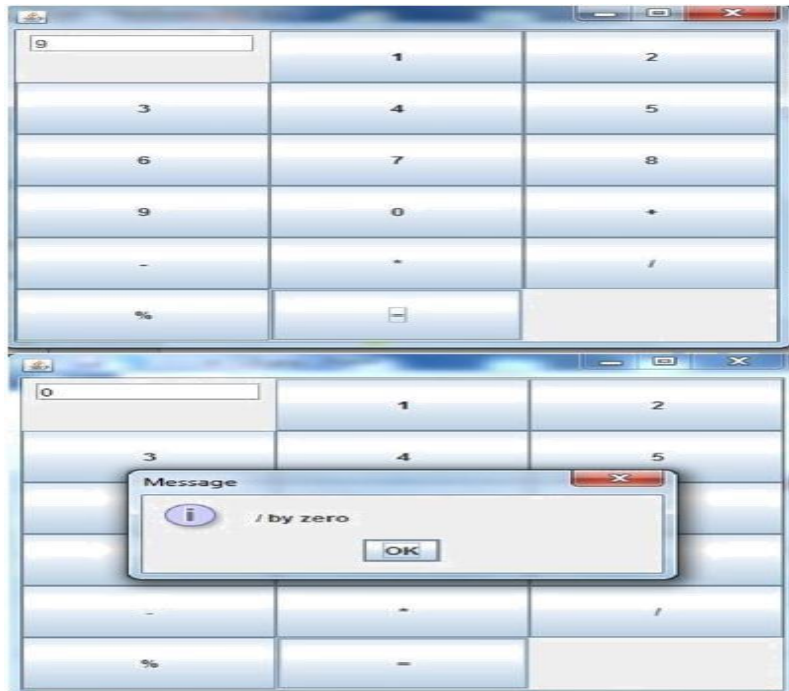


```
break;
case "0":
{
v = v + "0";
tf1.setText(v);
}
break;
case "+":
{
op = "+";
v1 = tf1.getText();
v = "";
}
break;
case "-":
{
op = "-";
v1 = tf1.getText();
v = "";
}
break;
case "*":
{
op = "*";
v1 = tf1.getText(); v = "";
}
break;
case "/":
{
op = "/";
v1 = tf1.getText();
v = "";
}
break;
case "%":
{
op = "%";
v1 = tf1.getText();
v = "";
}
break;
case "=":
{
switch (op)
{
case "+":
```

```
{
    v = tf1.getText();
    if (v.equals(""))
    {
        v = "0";
    }
    long i = Long.parseLong(v1) + Long.parseLong(v);
    tf1.setText(String.valueOf(i));
    v="";
}
break;
case "-":
{
    v = tf1.getText();
    if (v.equals(""))
    {
        v = "0";
    }
    long i = Long.parseLong(v1) - Long.parseLong(v);
    tf1.setText(String.valueOf(i));
    v="";
}
break;
case "*":
{
    v = tf1.getText();
    if (v.equals(""))
    {
        v = "0";
    }
    long i = Long.parseLong(v1) * Long.parseLong(v);
    tf1.setText(String.valueOf(i));
    v="";
}
break;
case "/":
{
    try
    {
        v = tf1.getText();
        if (v.equals(""))
        {
            v = "0";
        }
        long i = Long.parseLong(v1) / Long.parseLong(v);
```

```
        tf1.setText(String.valueOf(i));
        v="";
    }
    catch (Exception ex)
    {
        JOptionPane.showMessageDialog(this, ex.getMessage());
    }
    }
    break;
    case "% ":
    {
        try
        {
            v = tf1.getText();
            if (v.equals(""))
            {
                v = "0";
            }
            long i = Long.parseLong(v1) % Long.parseLong(v);
            tf1.setText(String.valueOf(i));
            v="";
        }
        catch (Exception ex)
        {
            JOptionPane.showMessageDialog(this, ex.getMessage());
        }
        }
        break;
    }
    break;
    }
    }
}

public class Calc
{
    public static void main(String[] args)throws Exception
    {
        A a = new A();
    }
}
```

Output:

EXPERIMENT NO: 03a**Name of the Program:**

Develop an Applet in java that displays a Simple Message

Source Code:

```
import java.applet.Applet;
import java.awt.*;
/*<applet code = "Sms.class" width="500" height="500">
</applet>*/
public class Sms extends Applet
{
    public void init()
    {
    }
    public void paint(Graphics g)
    {
        g.setColor(Color.blue);
        Font font = new Font("verdana", Font.BOLD, 15);
        g.setFont(font);
        g.drawString("Welcome To Malla Reddy Eng College",
        50, 50);
    }
}
```

Output:

EXPERIMENT NO: 03b**Name of the Program:**

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 button named "Compute" is clicked

Source Code:

```
import java.applet.Applet;

import java.awt.*;

import java.awt.event.*;

/*<applet code = "Fact.class" width="200" height="100">
</applet>*/

public class Fact extends Applet implements ActionListener

    {

        Label l1, l2, l3;

        TextField tf1, tf2;

        Button b1;

        public void init()

        {

            setSize(400, 200);

            FlowLayout g = new FlowLayout();

            setLayout(g);

            l1 = new Label("Enter Value");

            l1.setForeground(Color.BLUE);

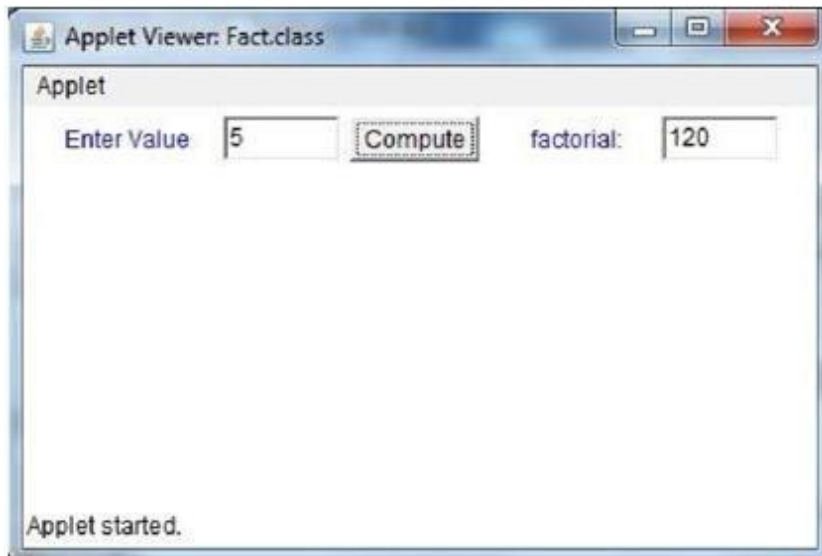
            add(l1);

            tf1 = new TextField(5);

            tf1.setText("0");

            add(tf1);
```

```
b1 = new Button("Compute");  
b1.addActionListener(this);  
add(b1);  
  
l3 = new Label();  
add(l3);  
  
l2 = new Label("factorial: ");  
l2.setForeground(Color.BLUE);  
add(l2);  
  
tf2 = new TextField(5); add(tf2);  
}  
  
public void actionPerformed(ActionEvent ae)  
{  
  
    long n = Integer.parseInt(tf1.getText());  
  
    long f = 1;  
    while (n != 0)  
    {  
        f = f * n;  
        n--;  
    }  
  
    tf2.setText(String.valueOf(f));  
}  
}
```

Output:

EXPERIMENT NO: 04**Name of the Program:**

Write a Program that creates User Interface to perform Integer Divisions. The user enters two numbers in 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 integers, the program would throw a NumberFormatException. If Num2 is Zero, and the program would throw an ArithmeticException. Display the Exception in message box.

Source Code:

```
import java.awt.*;

import java.awt.event.*;

import java.applet.Applet;

import javax.swing.*;

/*<applet code="AddEvent.class" height=500 width=500></applet>*/

public class AddEvent extends Applet implements ActionListener

{

    TextField tf1;

    TextField tf2;

    Button b;

    TextField tf3;

    Label l;

    public void init()

    {

        l=new Label("enter the numbers and press divide button");

        tf1=new TextField("",5);

        tf2=new TextField("",5);

        tf3=new TextField("",5);

        b=new Button("Divide");
```

```
add(l);

add(tf1);

add(tf2);

add(b);

add(tf3);

b.addActionListener(this);

}

public void actionPerformed(ActionEvent ae)

{

if(ae.getActionCommand()=="Divide")

{

try

{

int n1=Integer.parseInt(tf1.getText());

int n2=Integer.parseInt(tf2.getText());

int n=n1/n2;

tf3.setText(""+n);

}

catch(ArithmeticException e1)

{

JOptionPane.showMessageDialog(null,"Arithmetic Exception");

}

catch(NumberFormatException e2)

{

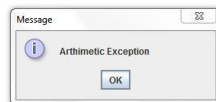
JOptionPane.showMessageDialog(null,"NumberFormatException");
```

}

}

}

}

Output:

EXPERIMENT NO: 05**Name of the Program:**

Write Java Program that implements a multithread application that has three threads. First thread generates random integer for every second and if the value is even, second thread computes the square of number and prints. If the value is odd, the third thread will print the value of cube of number.

Source Code:

```
import java.util.*;
class even implements Runnable
{
    public int x;
    publiceven(int x)
    {
        this.x = x;
    }
    public void run()
    {
        System.out.println("Thread Name:Even Thread and " + x + "is even Number and Square of " +
        x + " is: " + x * x);
    }
}
class odd implements Runnable
{
    public int x;
    public odd(int x)
    {
        this.x = x;
    }
    public void run()
    {
        System.out.println("Thread Name:ODD Thread and " + x + " is odd number and Cube of " + x
        + " is: " + x * x * x);
    }
}
class A extends Thread
{
    public String tname;
    public Random r;
    public Thread t1, t2;
    public A(String s)
    {
        tname = s;
    }
    public void run()
    {
        int num = 0;
```

```
r = new Random();
try
{
for (int i = 0; i < 50; i++)
{
num =r.nextInt(100);
System.out.println("Main Thread and Generated Number is " + num);
if (num % 2 == 0)
{
t1 = new Thread(new even(num));
t1.start();
}
else
{
t2 = new Thread(new odd(num));
t2.start();
}
Thread.sleep(1000);
System.out.println(".....");
}
}
catch (Exception ex)
{
System.out.println(ex.getMessage());
}
}
}
public class Mthread
{
public static void main(String[] args) throws Exception
{
A a = new A("One");
a.start();
}
}
```

Output:**Output - mthread (run)**

```
run:
Main Thread and Generated Number is 95
Thread Name:ODD Thread and 95 is odd number and Cube of 95 is: 857375
-----
Main Thread and Generated Number is 90
Thread Name:Even Thread and 90is even Number and Square of 90 is: 8100
-----
Main Thread and Generated Number is 35
Thread Name:ODD Thread and 35 is odd number and Cube of 35 is: 42875
-----
Main Thread and Generated Number is 41
Thread Name:ODD Thread and 41 is odd number and Cube of 41 is: 68921
-----
Main Thread and Generated Number is 75
Thread Name:ODD Thread and 75 is odd number and Cube of 75 is: 421875
-----
Main Thread and Generated Number is 29
Thread Name:ODD Thread and 29 is odd number and Cube of 29 is: 24389
-----
Main Thread and Generated Number is 55
Thread Name:ODD Thread and 55 is odd number and Cube of 55 is: 166375
-----
Main Thread and Generated Number is 84
Thread Name:Even Thread and 84is even Number and Square of 84 is: 7056
-----
Main Thread and Generated Number is 36
Thread Name:Even Thread and 36is even Number and Square of 36 is: 1296
-----
Main Thread and Generated Number is 51
Thread Name:ODD Thread and 51 is odd number and Cube of 51 is: 132651
-----
Main Thread and Generated Number is 15
Thread Name:ODD Thread and 15 is odd number and Cube of 15 is: 3375
```

EXPERIMENT NO: 06**Name of the Program:**

Write a java program for the following:

- i) Create a doubly linked list of elements.
- ii) Delete a given element from the above list.
- iii) Display the contents of the list after the deletion.

Source Code:

```
import java.util.Scanner;
class Node
{
    protected int data;
    protected Node next, prev;
    /* Constructor */
    public Node()
    {
        next = null;
        prev = null;
        data = 0;
    }
    /* Constructor */
    public Node(int d, Node n, Node p)
    {
        data = d;
        next = n;
        prev = p;
    }
    /* Function to set link to next node */
    public void setLinkNext(Node n)
    {
        next = n;
    }
    /* Function to set link to previous node */
    public void setLinkPrev(Node p)
    {
        prev = p;
    }
    /* Function to get link to next node */
    public Node getLinkNext()
    {
        return next;
    }
    /* Function to get link to previous node */
    public Node getLinkPrev()
    {

```

```
        return prev;
    }
    /* Function to set data to node */
    public void setData(int d)
    {
        data = d;
    }
    /* Function to get data from node */
    public int getData()
    {
        return data;
    }
}
/* Class linkedList */
class linkedList
{
    protected Node start;
    protected Node end ;
    public int size;
    /* Constructor */
    public linkedList()
    {
        start = null;
        end = null;
        size = 0;
    }
    /* Function to check if list is empty */
    public boolean isEmpty()
    {
        return start == null;
    }
    /* Function to get size of list */
    public int getSize()
    {
        return size;
    }
    /* Function to insert element at beginning */
    public void insertAtStart(int val)
    {
        Node nptr = new Node(val, null, null);
        if(start == null)
        {
            start = nptr;
            end = start;
        }
        else
```



```
{
    start.setLinkPrev(nptr);
    nptr.setLinkNext(start);
    start = nptr;
}
size++;
}

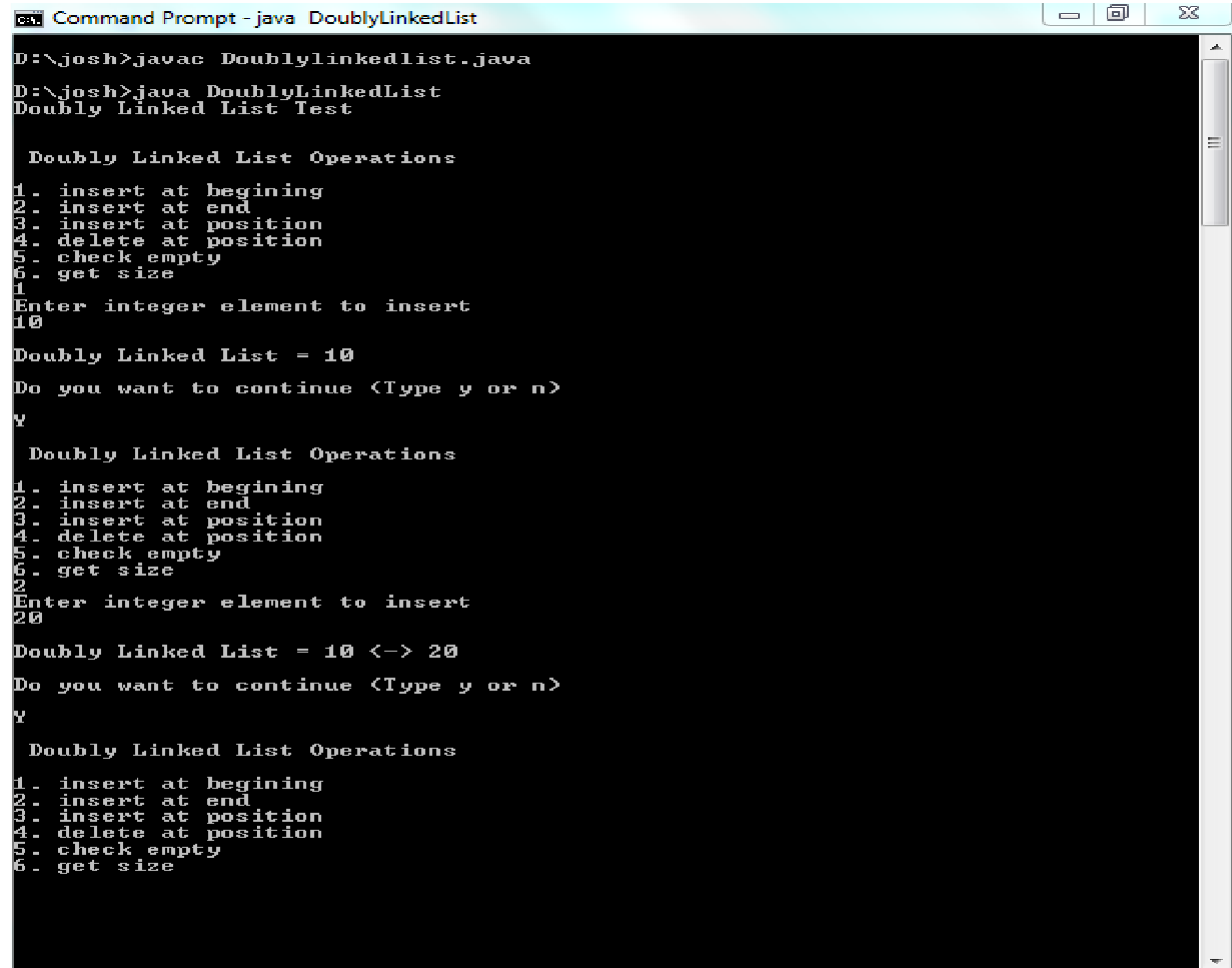
/* Function to insert element at end */
public void insertAtEnd(int val)
{
    Node nptr = new Node(val, null, null);
    if(start == null)
    {
        start = nptr;
        end = start;
    }
    else
    {
        nptr.setLinkPrev(end);
        end.setLinkNext(nptr);
        end = nptr;
    }
    size++;
}

/* Function to insert element at position */
public void insertAtPos(int val , int pos)
{
    Node nptr = new Node(val, null, null);
    if (pos == 1)
    {
        insertAtStart(val);
        return;
    }
    Node ptr = start;
    for (int i = 2; i <= size; i++)
    {
        if (i == pos)
        {
            Node tmp = ptr.getLinkNext();
            ptr.setLinkNext(nptr);
            nptr.setLinkPrev(ptr);
            nptr.setLinkNext(tmp);
            tmp.setLinkPrev(nptr);
        }
        ptr = ptr.getLinkNext();
    }
}
```

```
size++ ;
}
/* Function to delete node at position */
public void deleteAtPos(int pos)
{
    if (pos == 1)
    {
        if (size == 1)
        {
            start = null;
            end = null;
            size = 0;
            return;
        }
        start = start.getLinkNext();
        start.setLinkPrev(null);
        size--;
        return ;
    }
    if (pos == size)
    {
        end = end.getLinkPrev();
        end.setLinkNext(null);
        size-- ;
    }
    Node ptr = start.getLinkNext();
    for (int i = 2; i <= size; i++)
    {
        if (i == pos)
        {
            Node p = ptr.getLinkPrev();
            Node n = ptr.getLinkNext();
            p.setLinkNext(n);
            n.setLinkPrev(p);
            size-- ;
            return;
        }
        ptr = ptr.getLinkNext();
    }
}
/* Function to display status of list */
public void display()
{
    System.out.print("\nDoubly Linked List = ");
    if (size == 0)
    {
```

```
        System.out.print("empty\n");
        return;
    }
    if (start.getLinkNext() == null)
    {
        System.out.println(start.getData() );
        return;
    }
    Node ptr = start;
    System.out.print(start.getData()+ " <-> ");
    ptr = start.getLinkNext();
    while (ptr.getLinkNext() != null)
    {
        System.out.print(ptr.getData()+ " <-> ");
        ptr = ptr.getLinkNext();
    }
    System.out.print(ptr.getData()+ "\n");
}
}
/* Class DoublyLinkedList */
public class DoublyLinkedList
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        /* Creating object of linkedList */
        linkedList list = new linkedList();
        System.out.println("Doubly Linked List Test\n");
        char ch;
        /* Perform list operations */
        do
        {
            System.out.println("\n Doubly Linked List Operations\n");
            System.out.println("1. insert at begining");
            System.out.println("2. insert at end");
            System.out.println("3. insert at position");
            System.out.println("4. delete at position");
            System.out.println("5. check empty");
            System.out.println("6. get size");
            int choice = scan.nextInt();
            switch (choice)
            {
                case 1 :
                    System.out.println("Enter integer element to insert");
                    list.insertAtStart( scan.nextInt() );
                    break;
```

```
case 2 :
System.out.println("Enter integer element to insert");
list.insertAtEnd( scan.nextInt() );
break;
case 3 :
System.out.println("Enter integer element to insert");
int num = scan.nextInt() ;
System.out.println("Enter position");
int pos = scan.nextInt() ;
if (pos < 1 || pos > list.getSize() )
System.out.println("Invalid position\n");
else
list.insertAtPos(num, pos);
break;
case 4 :
System.out.println("Enter position");
int p = scan.nextInt() ;
if (p < 1 || p > list.getSize() )
System.out.println("Invalid position\n");
else
list.deleteAtPos(p);
break;
case 5 :
System.out.println("Empty status = "+ list.isEmpty());
break;
case 6 :
System.out.println("Size = "+ list.getSize() +" \n");
break;
default :
System.out.println("Wrong Entry \n ");
break;
}
/* Display List */
list.display();
System.out.println("\nDo you want to continue (Type y or n) \n");
ch = scan.next().charAt(0);
}
while (ch == 'Y' || ch == 'y');
}
}
```

Output:

```
Command Prompt - java DoublyLinkedList
D:\josh>javac DoublyLinkedList.java
D:\josh>java DoublyLinkedList
Doubly Linked List Test

Doubly Linked List Operations
1. insert at begining
2. insert at end
3. insert at position
4. delete at position
5. check empty
6. get size
1
Enter integer element to insert
10
Doubly Linked List = 10
Do you want to continue <Type y or n>
y

Doubly Linked List Operations
1. insert at begining
2. insert at end
3. insert at position
4. delete at position
5. check empty
6. get size
2
Enter integer element to insert
20
Doubly Linked List = 10 <-> 20
Do you want to continue <Type y or n>
y

Doubly Linked List Operations
1. insert at begining
2. insert at end
3. insert at position
4. delete at position
5. check empty
6. get size
```

EXPERIMENT NO: 07**Name of the Program:**

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 radio button, an appropriate message with “stop” or “Ready” or “GO” should appear above the button in selected color. Initially, there is no message shown.

Source Code:

```
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;
import java.awt.event.*;
class A extends JFrame implements ItemListener
{
    public JLabel l1, l2;
    public JRadioButton r1,
    r2, r3; public
    ButtonGroup bg; public
    JPanel p, p1;
    public A()
    {
        setDefaultCloseOperation(JFrame.EXIT_ON_C
        LOSE); setLayout(new GridLayout(2, 1));
        setSize(800, 400);
        p = new JPanel(new FlowLayout());
        p1 = new JPanel(new FlowLayout());
        l1 = new JLabel();
        Font f = new Font("Verdana",
        Font.BOLD, 60); l1.setFont(f);
        add(l1);
        p.add(l1);
        add(p);
        l2 = new JLabel("Select
        Lights"); p1.add(l2);
        JRadioButton r1 = new JRadioButton("Red
        Light"); r1.setBackground(Color.red);
        p1.add(r1);
        r1.addItemListener(this);
        JRadioButton r2 = new JRadioButton("Yellow
        Light"); r2.setBackground(Color.YELLOW);
        p1.add(r2);
        r2.addItemListener(this);
        JRadioButton r3 = new JRadioButton("Green
        Light"); r3.setBackground(Color.GREEN);
        p1.add(r3);
        r3.addItemListener(this);
    }
}
```

```
add(p1);
bg = new
ButtonGroup();
bg.add(r1);
bg.add(r2);
bg.add(r3);
setVisible(true);
}
public void itemStateChanged(ItemEvent i)
{
JRadioButton jb = (JRadioButton) i.getSource(); switch (jb.getText())
{
case "Red Light":
{
l1.setText("STOP"); l1.setForeground(Color.red);
}
break;
case "Yellow Light":
{
l1.setText("Ready");
l1.setForeground(Color.YELLOW);
}
break;
case "Green Light":
{
l1.setText("GO");
l1.setForeground(Color.GREEN);
}
break;
}
}
}
public class TLight
{
public static void main(String[] args)
{
A a = new A();
}
}
```

OUTPUT:

EXPERIMENT NO: 08**Name of the Program:**

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 subclass that each one of the classes extends the Class Shape. Each one of the classes contains only the method printArea () that prints the area of Shape.

Source Code:

```
abstract class shape
{
    public int x, y;
    public abstract void printArea();
}
class Rectangle extends shape
{
    public void printArea()
    {
        System.out.println("Area of Rectangle is " + x * y);
    }
}
class Triangle extends shape
{
    public void printArea()
    {
        System.out.println("Area of Triangle is " + (x * y) / 2);
    }
}
class Circle extends shape
{
    public void printArea()
    {
        System.out.println("Area of Circle is " + (22 * x * x) / 7);
    }
}
public class Abstex
{
    /* @param args the command line arguments*/
    public static void main(String[] args) {
        // TODO code application logic here
        Rectangle r = new Rectangle();
        r.x = 10;
        r.y = 20;
        r.printArea();
        System.out.println(".....");
        Triangle t = new Triangle();
    }
}
```

```
t.x = 30;  
t.y = 35;  
t.printArea();  
System.out.println(".....");  
Circle c = new Circle();  
c.x = 2;  
c.printArea();  
System.out.println(".....");  
}  
}
```

Output:

```
run:  
Area of Rectangle is 200  
-----  
Area of Triangle is 525  
-----  
Area of Circle is 12  
-----  
BUILD SUCCESSFUL (total time: 3 seconds)  
|
```

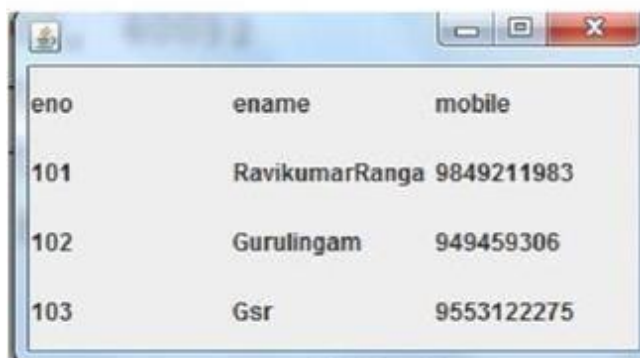
EXPERIMENT NO: 09**Name of the Program:**

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 rows in table. The elements are separated by commas. Write java program to display the table using Label in Grid Layout.

Source Code:

```
import java.io.*;
import java.util.*;
import java.awt.*;
import javax.swing.*;
import javax.swing.event.*;
class A extends JFrame
{
    public A()
    {
        setSize(400, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CL
        OSE);
        GridLayout g = new GridLayout(0, 3);
        setLayout(g);
        try
        {
            FileInputStream fin= new FileInputStream("D:\\emp.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 Tbl
```

```
{  
    public static void main(String[]  
        args)  
    {  
        A a = new A();  
    }  
}
```

Output:

eno	ename	mobile
101	RavikumarRanga	9849211983
102	Gurulingam	949459306
103	Gsr	9553122275

EXPERIMENT NO: 10**Name of the Program:**

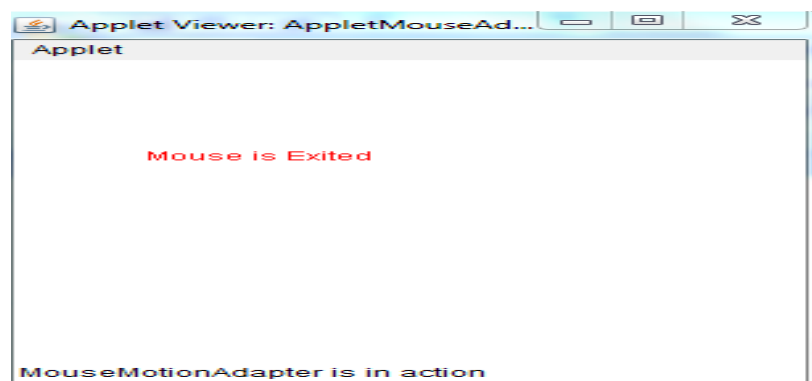
Write a Java Program that handles all mouse events and show event name at the center of the window when the mouse event is fired.(Use Adapter Classes)

Source Code:

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
/*<applet code="AppletMouseAdapterDemo.class" width="300" height="300">
</applet>*/
public class AppletMouseAdapterDemo extends Applet
{
    String str="";
    public void init()
    {
        AdapterDemo ad= new AdapterDemo(this);
        addMouseListener(ad);
    }
    public void paint(Graphics g)
    {
        g.drawString(str, 50, 100);
        showStatus("MouseMotionAdapter is in action");
    }
}
class AdapterDemo extends MouseAdapter
{
    AppletMouseAdapterDemo amad2;
    public AdapterDemo(AppletMouseAdapterDemo amad1)
    {
        amad2 = amad1;
    }
    public void mouseClicked(MouseEvent e)
    {
        amad2.str = "Mouse is clicked";
        amad2.setForeground(Color.red);
        amad2.repaint();
    }
    public void mouseEntered(MouseEvent e)
    {
        amad2.str = "Mouse is Entered";
        amad2.setForeground(Color.blue);
        amad2.repaint();
    }
    public void mouseMoved(MouseEvent e)
    {

```

```
amad2.str = "Mouse is moved";
amad2.setForeground(Color.red);
amad2.repaint();
}
public void mouseExited(MouseEvent e)
{
    amad2.str = "Mouse is Exited";
    amad2.setForeground(Color.red);
    amad2.repaint();
}
public void mouseDragged(MouseEvent e)
{
    amad2.str = "Mouse is dragged";
    amad2.setForeground(Color.red);
    amad2.repaint();
}
}
```

Output:

EXPERIMENT NO: 11**Name of the Program:**

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

Source Code:

```
import java.util.*;
import java.io.*;
public class Hashtbl
{
    public static void main(String[] args) {
        try
        {
            FileInputStream fs = new FileInputStream("D:\\ph.txt");
            Scanner sc = new Scanner(fs).useDelimiter("\\s+");
            Hashtable<String, String> ht = new Hashtable<String,
            String>();
            String[] arrayList;
            String a;
            System.out.println("Welcome TO Anurag Eng
            College"); System.out.println("HASH TABLE IS");
            System.out.println(" ----- ");
            System.out.println("KEY : VALUE");
            while (sc.hasNext())
            {
                a= sc.nextLine();
                arrayList = a.split("\\s+");
                ht.put(arrayList[0], arrayList[1]);
                System.out.println(arrayList[0] + ":" + arrayList[1]);
            }
            System.out.println("Welcome TO Anurag Eng
            College"); System.out.println("----MENU----- ");
            System.out.println("----1.Search by Name----- ");
            System.out.println("----2.Search by Mobile ----- ");
            System.out.println("----3.Exit ----- ");
            String opt = "";
            String name, mobile;
            Scanner s = new Scanner(System.in);
            while (opt != "3")
            {
                System.out.println("Enter Your Option 1,2,3");
                opt = s.next();
                switch (opt)
                {
                    case "1":
```

```
{
System.out.println("Enter Name");
name = s.next();
if (ht.containsKey(name))
{
System.out.println("Mobile is " + ht.get(name));
}
else
{
System.out.println("Not Found");
}
}
break;
case "2":
{
System.out.println("Enter mobile");
mobile = s.next();
if (ht.containsValue(mobile))
{
for (Map.Entry e : ht.entrySet())
{
if (mobile.equals(e.getValue()))
{
System.out.println("Name is " + e.getKey());
}
}
}
else
{
System.out.println("Not Found");
}
}
break;
case "3":
{
opt = "3";
System.out.println("Menu Successfully Exited");
}
break;
default:
System.out.println("Choose Option between 1 and Three");
break;
}
}
}
catch (Exception ex)
{
}
```



```
System.out.println(ex.getMessage());  
}  
}  
}
```

Output:

```
ph.txt - Notepad  
File Edit Format View Help  
RavikumarRanga 9849211983  
Gurulingam 949459306  
Gsr 9553122275
```



```
Output - hashtbl (run)  
run:  
HASH TABLE IS  
-----  
KEY      :  VALUE  
RavikumarRanga:9849211983  
Gurulingam:949459306  
Gsr:9553122275  
  
----MENU-----  
----1.Search by Name-----  
----2.Search by Mobile-----  
----3.Exit-----  
Enter Your Option 1,2,3  
1  
Enter Name  
RavikumarRanga  
Mobile is 9849211983  
Enter Your Option 1,2,3  
2  
Enter mobile  
9553122275  
Name is Gsr  
Enter Your Option 1,2,3  
3  
Menu Successfully Exited  
BUILD SUCCESSFUL (total time: 32 seconds)  
|
```

EXPERIMENT NO: 12**Name of the Program:**

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

Source Code:

```
import java.lang.*;
import java.lang.Thread;
class Q
{
    int n;
    boolean valueSet=false;
    synchronized int get()
    {
        while(!valueSet)
        try{
            wait();
        }
        catch(InterruptedException e)
        {
            System.out.println("InterruptedException catch");
        }
        System.out.println("got:"+n);
        valueSet=false;
        notify();
        return n;
    }
    synchronized void put(int n)
    {
        while(valueSet)
        try{
            wait();
        }
        catch(InterruptedException e)
        {
            System.out.println("InterruptedException catch");
        }
        this.n=n;
        valueSet=true;
        System.out.println("put:"+n);
        notify();
    }
}
class Producer implements Runnable
{
    Q q;
```

```
    Producer(Q q)
    {
        this.q=q;
        new Thread(this,"producer").start();
    }
    public void run()
    {
        int i=0;
        while(true)
        {
            q.put(i++);
        }
    }
}
class Consumer implements Runnable
{
    Q q;
    Consumer(Q q)
    {
        this.q=q;
        new Thread(this,"consumer").start();
    }
    public void run()
    {
        while(true)
        {
            q.get();
        }
    }
}
class PC
{
public static void main(String args[])
    {
        Q q=new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("press ctrl+c to stop");
    }
}
```

OUTPUT

cmd Command Prompt

```
put:29892
got:29892
put:29893
got:29893
put:29894
got:29894
put:29895
got:29895
put:29896
got:29896
put:29897
got:29897
put:29898
got:29898
put:29899
got:29899
put:29900
got:29900
put:29901
got:29901
put:29902
got:29902
put:29903
got:29903
D:\josh>
```

EXPERIMENT NO: 13**Name of the Program:**

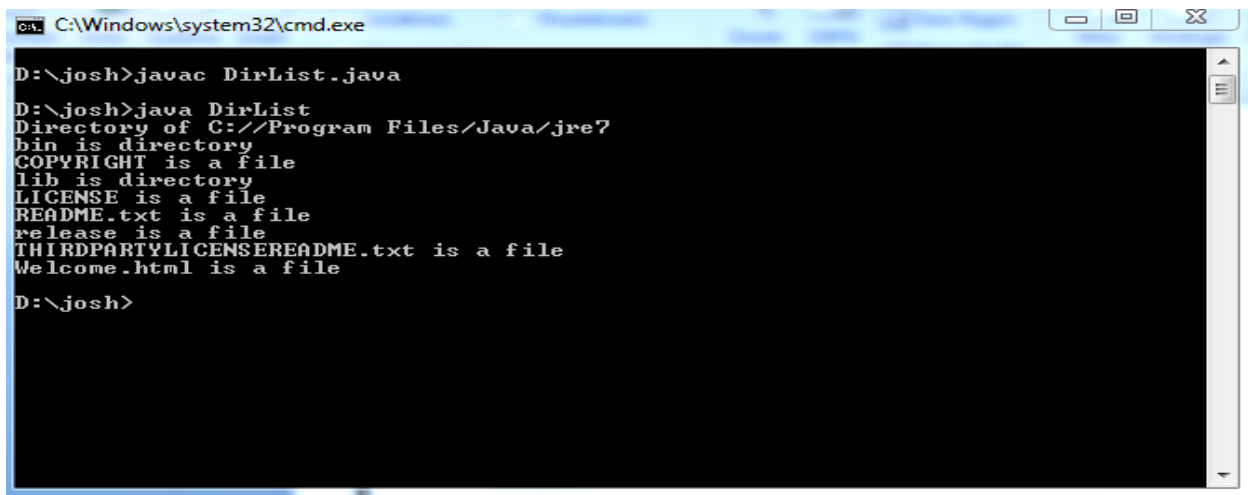
Write a Java program to list all the files in a directory including the files present in all its subdirectories.

Source Code:

```
import java.io.File;

class DirList
{
    public static void main(String args[])
    {
        String dirname = "C://Program Files/Java/jre7";
        File f1 = new File(dirname);
        if(f1.isDirectory())
        {
            System.out.println("Directory of " + dirname);
            String s[] = f1.list();
            for(int i=0; i< s.length;i++)
            {
                File f = new File(dirname + "/" +s[i]);
                if(f.isDirectory())
                {
                    System.out.println(s[i] + " is directory");
                }
                else
                {
                    System.out.println(s[i] + " is a file");
                }
            }
        }
    }
}
```

```
}  
else  
{  
    System.out.println(dirname + "is not a directory");  
}  
}  
}
```

Output:

The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The user is at the prompt "D:\josh>". They enter the command "javac DirList.java". The prompt changes to "D:\josh>". They then enter "java DirList". The program outputs the following text:
Directory of C://Program Files/Java/jre7
bin is directory
COPYRIGHT is a file
lib is directory
LICENSE is a file
README.txt is a file
release is a file
THIRDPARTYLICENSEREADME.txt is a file
Welcome.html is a file
The prompt returns to "D:\josh>".

EXPERIMENT NO: 14**Name of the Program:**

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

Source Code:

```
import java.util.Scanner;
```

```
public class QuickSort
```

```
{
```

```
    public static void sort(int[] arr)
```

```
{
```

```
    quickSort(arr, 0, arr.length- 1);
```

```
}
```

```
    public static void quickSort(int arr[], int low, int high)
```

```
{
```

```
    int i = low, j = high;
```

```
    int temp;
```

```
    int pivot = arr[(low + high) / 2];
```

```
    while (i <= j)
```

```
{
```

```
        while (arr[i] < pivot)
```

```
            i++;
```

```
        while (arr[j] > pivot)
```

```
            j--;
```

```
        if (i <= j)
```

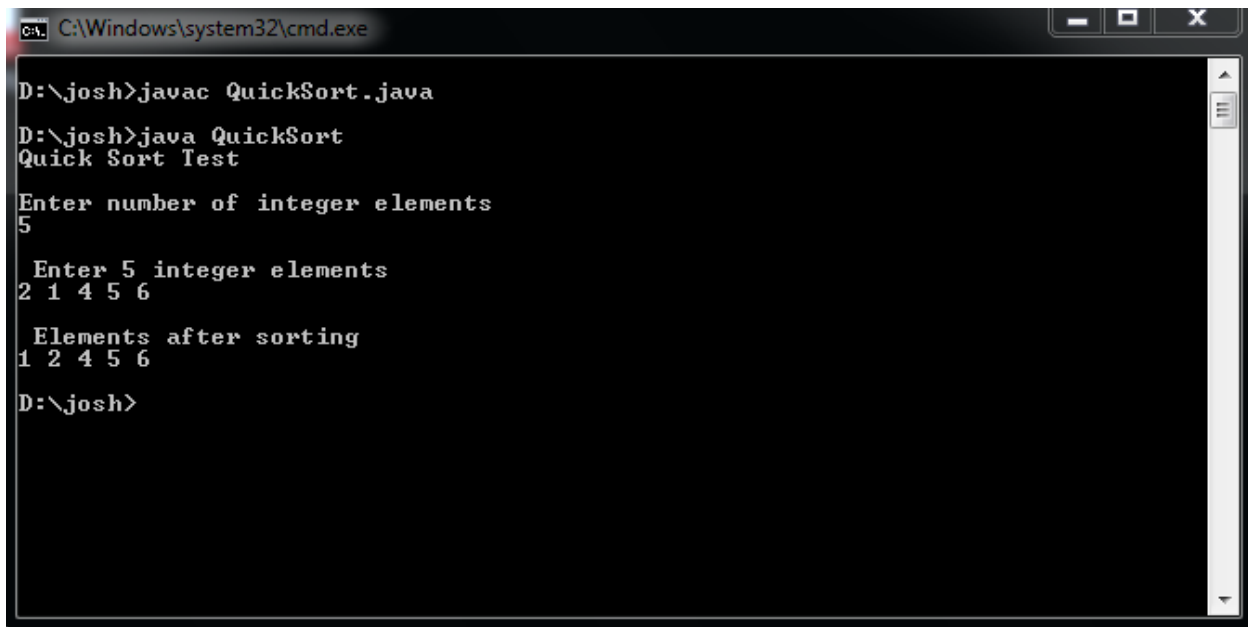
```
{
```

```
            temp = arr[i];
```

```
            arr[i] = arr[j];
```

```
            arr[j] = temp;
```

```
i++;  
j--;  
}}  
if (low < j)  
    quickSort(arr, low, j);  
if (i < high)  
    quickSort(arr, i, high);  
}  
  
public static void main(String[] args)  
{  
    Scanner scan = new Scanner( System.in );  
    System.out.println("Quick Sort Test\n");  
    int n, i;  
    System.out.println("Enter number of integer elements");  
    n = scan.nextInt();  
    int arr[] = new int[ n ];  
    System.out.println("\n Enter "+ n +" integer elements");  
    for (i = 0; i < n; i++)  
        arr[i] = scan.nextInt();  
    sort(arr);  
    System.out.println("\n Elements after sorting ");  
    for (i = 0; i < n; i++)  
        System.out.print(arr[i]+" ");  
    System.out.println();  
}  
}
```


OUTPUT:

```
C:\Windows\system32\cmd.exe

D:\josh>javac QuickSort.java
D:\josh>java QuickSort
Quick Sort Test
Enter number of integer elements
5
    Enter 5 integer elements
2 1 4 5 6
Elements after sorting
1 2 4 5 6
D:\josh>
```

EXPERIMENT NO: 15**Name of the Program:**

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:

```
public class MyBubbleSort

{

    // logic to sort the elements

    public static void bubble_srt(int array[])
    {
        int n = array.length;
        int k;
        for (int m = n; m >= 0; m--)
        {
            for (int i = 0; i < n - 1; i++)
            {
                k = i + 1;
                if (array[i] < array[k])
                {
                    swapNumbers(i, k, array);
                }
            }
            printNumbers(array);
        }
        private static void swapNumbers(int i, int j, int[] array)
        {
            int temp;
            temp = array[i];
            array[i] = array[j];
            array[j] = temp;
        }
        private static void printNumbers(int[] input)
        {
            for (int i = 0; i < input.length; i++)
            {
                System.out.print(input[i] + ", ");
            }
        }
    }
}
```

```
System.out.println("\n");
}
public static void main(String[] args)
{
int[] input = { 4, 2, 9, 6, 0, 1 };
bubble_srt(input);
}
}
```

OUTPUT



```
C:\Windows\system32\cmd.exe

D:\josh>javac MyBubbleSort.java

D:\josh>java MyBubbleSort
4, 9, 6, 2, 1, 0,
9, 6, 4, 2, 1, 0,
9, 6, 4, 2, 1, 0,
9, 6, 4, 2, 1, 0,
9, 6, 4, 2, 1, 0,
9, 6, 4, 2, 1, 0,
9, 6, 4, 2, 1, 0,
D:\josh>
```