

**LAB MANUAL
ON
JAVA PROGRAMMING**



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Teegala Krishna Reddy Engineering College

(Affiliated to JNTUH, Approved by AICTE, Accredited by NBA & NAAC 'A' Grade)

Medbowli, Meerpet (V), Balapur (M), Hyderabad -500097.

Ph: 9393959597, Email: infotkrec@gmail.com,

CS408PC: JAVA PROGRAMMING LAB

Course Objectives

- To write programs using abstract classes.
- To write programs for solving real world problems using java collection framework.
- To write multithreaded programs.
- To write GUI programs using swing controls in Java.
- To introduce java compiler and eclipse platform.
- To impart hands on experience with java programming.

Course Outcomes

- Able to write programs for solving real world problems using java collection framework.
- Able to write programs using abstract classes.
- Able to write multithreaded programs.
- Able to write GUI programs using swing controls in Java.

CS408PC: JAVA PROGRAMMING LAB

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.	5-6
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.	7-12
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.	13-16
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.	16-19
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.	19-21
6	Write a Java program for the following: a) Create a doubly linked list of elements. b) Delete a given element from the above list. c) Display the contents of the list after deletion.	22-26
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.	27-30
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	30-32

CS408PC: JAVA PROGRAMMING LAB

	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.	32-34
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).	34-36
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).	37-40
12	Write a Java program that correctly implements the producer – consumer problem using the concept of inter thread communication.	40-43
13	Write a Java program to list all the files in a directory including the files present in all its subdirectories.	43-47
14	Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order	47-49
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.	49-51

CS408PC: JAVA PROGRAMMING LAB

1.1 OBJECTIVE:

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 RESOURCES:

JDK1.8.0 is required, 1GB RAM, Hard Disk 80 GB

1.3 PROGRAM LOGIC:

1. Start the program
2. Create a class with the name Testprj
3. Read n values from keyboard using a Scanner class.
4. Find the (n % i == 0) using dynamic initialization
5. Check the (n % i == 0) value using if condition.

1.3 PROGRAM LOGIC:

Input: no input required.

Output: Displays all the even numbers from 1 to 20 in the console.

1. Start the program
2. Create a class with the name Testprj
3. Read n values from keyboard using a Scanner class.
4. Find the (n % i == 0) using dynamic initialization
5. Check the (n % i == 0) value using if condition.

1.5 PROCEDURE:

To execute a java program we require setting a class path:

1. C:\set path= C:\Program Files\Java\jdk1.8.0\bin;.;
2. C:\javac Testprj.java
3. C:\java Testprj

1.6 SOURCE CODE:

```
public class TestPrj
{
    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)
            {
```

CS408PC: JAVA PROGRAMMING LAB

```
System.out.print("\t "+i);  
    }  
    }  
}  
}
```

OUTPUT:

D:\TKREC\Java>javac TestPrj.java

D:\TKREC\Java>java TestPrj

Prog.is showing even no

2 4 6 8 10 12 14 16 18 20

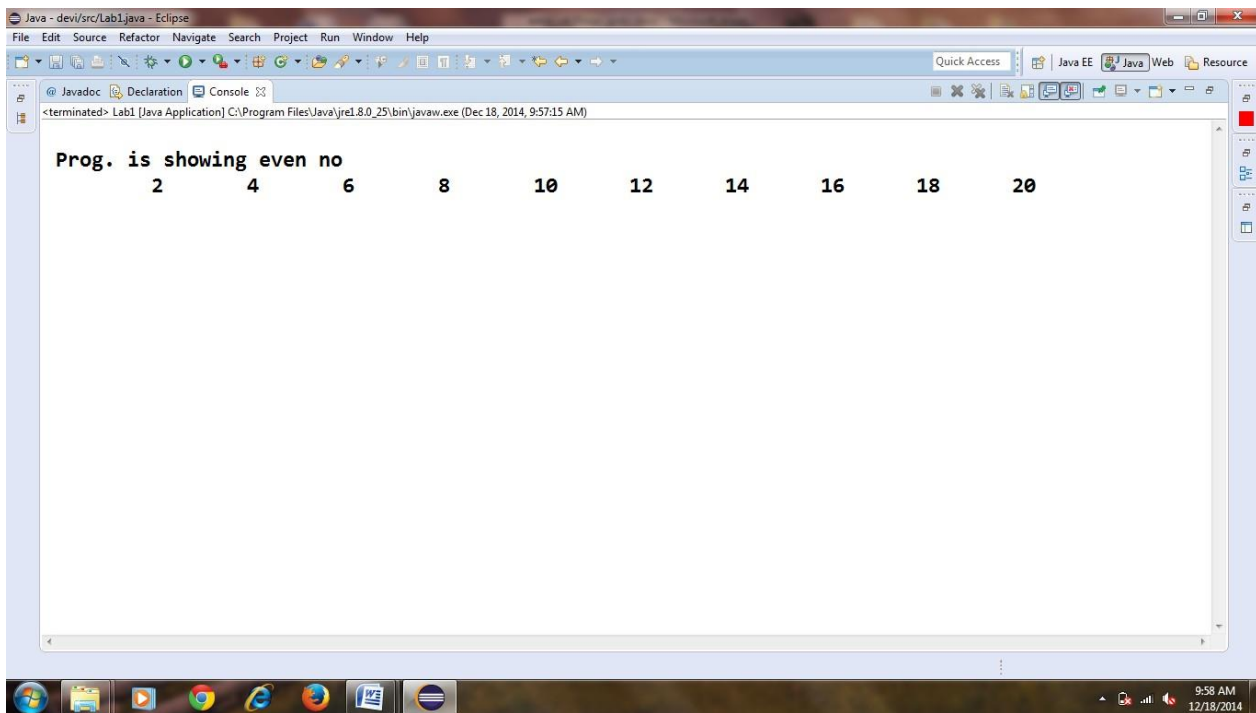


Figure 1: Output Console

Conclusion: Student will be able to learn about eclipse IDE.

Viva Questions for Program-1:

1. What is class?
2. What is the use of main method?
3. What is System?
4. What is out?
5. What is eclipse?

CS408PC: JAVA PROGRAMMING LAB

2.1 OBJECTIVE:

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

2.2 RESOURCES:

JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

2.3 PROGRAM LOGIC:

Input: Press the values and operation in the calculator

Output: Displays the values in the text field

1. Start the program
2. Create a class with the name "CalculatorPrg " extends JFrame implements ActionListener.
3. Declare the JButton ,JTextField ,JPanel variables.
4. Add listener to each button and to the layout.
5. Add Text Field to display the result.
6. Handle any Exceptions like divide by zero.

2.4 PROCEDURE:

To execute a java program we require setting a class path:

1. C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.;
2. C:\javac Calculator.java
3. C:\java Calculator

2.5 SOURCE CODE:

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

```
class CalculatorPrg extends JFrame implements ActionListener
{
    JButton b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12, b13, b14, b15, b16;
    JTextField tf1;
    JPanel p;
    String v = "";
    String v1 = "0";
    String op = "";

    CalculatorPrg()
    {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 400);
        p = new JPanel(new FlowLayout());
        tf1 = new JTextField(10);
```

CS408PC: JAVA PROGRAMMING LAB

```
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);
```


CS408PC: JAVA PROGRAMMING LAB

```
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;
    }
}
```

CS408PC: JAVA PROGRAMMING LAB

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

CS408PC: JAVA PROGRAMMING LAB

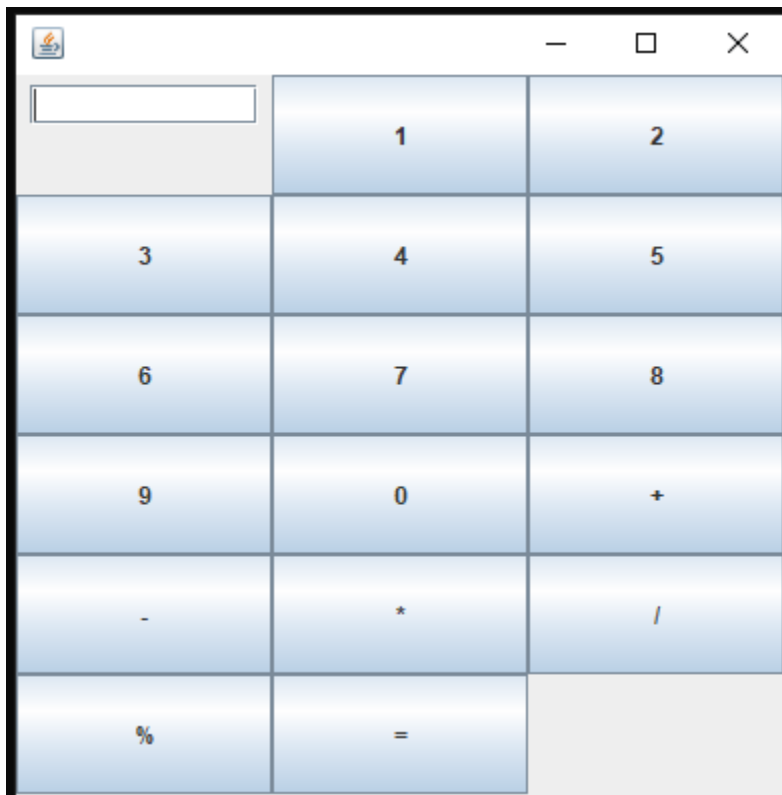
```
break;
case "%": {          v = tf1.getText();          if (v.equals("")) {
v = "0";
}
long i = Long.parseLong(v1) % Long.parseLong(v);
tf1.setText(String.valueOf(i));
v="";
}
break;
}}
break;
} } }
```

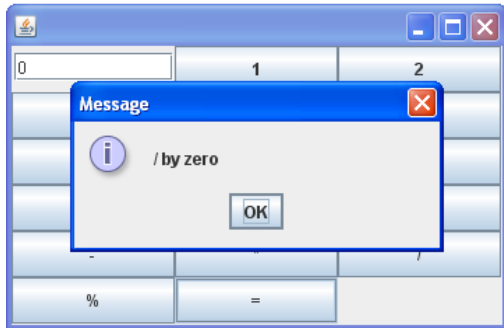
```
public class CalculatorTest
{
    public static void main(String[] args)
    {
        CalculatorPrg obj = new CalculatorPrg();
    }
}
```

OUTPUT:

D:\TKREC\Java>javac CalculatorTest.java

D:\TKREC\Java>java CalculatorTest





Conclusion: Student will learn the concept about applet and java.awt package

Viva Questions for Program 2:

1. Where are the applets executed?
2. Which tag is used to embed an applet into a HTML page?
3. Where are the borders available in swing?
4. What is Hot Java?
5. What is an adapter class?

3 a) To Develop an Applet to Display Simple Message

3.1 OBJECTIVE:

To develop an Applet in java that displays a Simple Message

3.2 RESOURCES:

JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

3.3 PROGRAM LOGIC:

1. Start the program
2. Create a class with the name Simple extends Applet.
3. Declare the Font class object.
4. Set the font.
5. Call the setFont method and drawString method g.setFont(font);
6. Create the HTML file for applet tag.

3.4 PROCEDURE:

To execute a java program we require setting a class path:

- 1.C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.;
- 2.C:\javac Simple.java
- 3.C:\appletviewer Simple.java

3.5 SOURCE CODE:

CS408PC: JAVA PROGRAMMING LAB

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

public class Simple extends Applet
{
    public void init()
    {
        public void paint(Graphics g)
        {
            g.setColor(Color.blue);
            Font font = new Font("verdana", Font.BOLD, 12);
            g.setFont(font);
            g.drawString("Welcome To TKR Eng College",40,50);
        }
    }
}
```

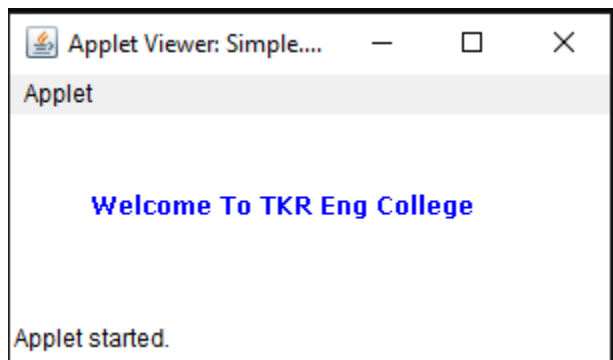
Simple.html

```
<html>
<body>
<applet code="Simple.class" width="300" height="100">
</applet>
</body>
</html>
```

This page will compile the java file and execute html file through applet viewer

OUTPUT:

```
D:\TKREC\Java>javac Simple.java
D:\TKREC\Java> appletviewer Simple.html
```



Conclusion: Student will learn how to display output through Applet.

3 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. To Develop an Applet to Display Simple Message

3.1 OBJECTIVE:

Develop an Applet that receives an integer in one text field & compute its factorial value & returns it in another text filed when the button “Compute” is clicked.

3.2 RESOURCES:

JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

3.3 PROGRAM LOGIC:

Input: Value of variable.

Output: Compute the factorial value

1. Start the program
2. Create a class with the name „Fact“ extends Applet which implements ActionListener.
3. Declare the Label,TextField, Button variables.
4. Call init method
- 5 Call the setLayout method, setLayout(g);
6. Create the HTML file for applet tag.

3.4 PROCEDURE:

To execute a java program we require setting a class path:

1. C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.;
2. C:\javac Fact.java
3. C:\appletviewer Fact.html

3.5 SOURCE CODE:

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

```
public class Fact extends Applet implements ActionListener  
{  
    Label l1, l2, l3;  
    TextField tf1, tf2;  
    Button b1;  
    public void init()  
    {  
        setSize(300, 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));
}
}
```

Fact.html

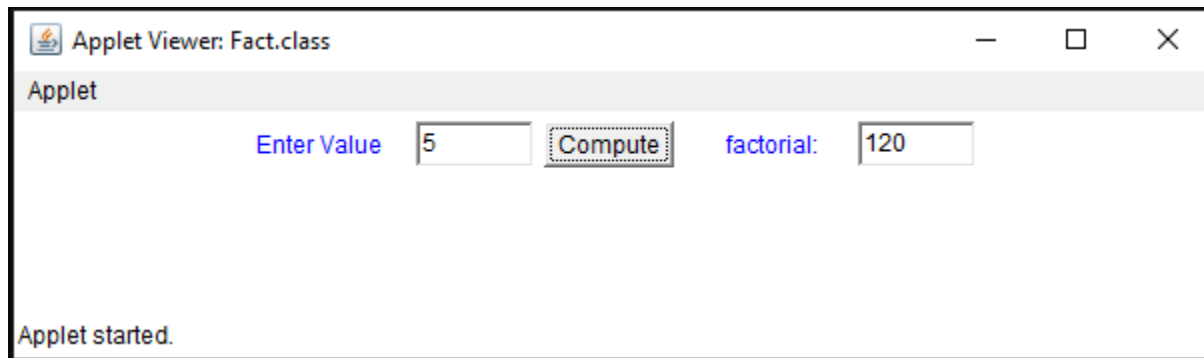
```
<html>
<body>
<applet code="Fact.class" width="300" height="100">
</applet>
</body>
</html>
```

OUTPUT:

D:\TKREC\Java>javac javac Fact.java

D:\TKREC\Java>java appletviewer fact.html

CS408PC: JAVA PROGRAMMING LAB



Conclusion: Student will learn about the action events in java.awt package

Viva Questions for Program 3:

1. What is the difference between a window and a frame?
2. Which model is used to provide actions to AWT components?
3. What is the default layout in an applet?
4. What is the default layout in a frame?
5. What is an applet?

4.1 OBJECTIVE:

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.

4.2 RESOURCES:

JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

4.3 PROGRAM LOGIC:

Input: Values of Text field T1, T2

Output: Displays the result in Text field T3

1. Start the program
2. Create a class with the name „A extends JFrame“.
3. Declare the JLabel,JTextField,JButton variables.
4. Declare the default constructor of the class.
5. Add buttons,TextField and labels to the Flow Layout.
6. Call ActionPerformed method.

4.4 PROCEDURE:

To execute a java program we require setting a class path:

1. C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.
2. C:\javac IntDivApp.java

3. C:\java IntDivApp

4.5 SOURCE CODE:

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

class IntDiv extends JFrame implements ActionListener
{
    JLabel l1, l2, l3;
    JTextField tf1, tf2, tf3;
    JButton b1;

    IntDiv()
    {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new FlowLayout());
        l1 = new JLabel("Welcome");
        setSize(800, 200);
        l1 = new JLabel("Enter Number1");
        add(l1);
        tf1 = new JTextField(10);
        add(tf1);
        l2 = new JLabel("Enter Number2");
        add(l2);
        tf2 = new JTextField(10);
        add(tf2);
        l3 = new JLabel("Result");
        add(l3);
        tf3 = new JTextField(10);
        add(tf3);
        b1 = new JButton("Divide");
        add(b1);
        b1.addActionListener(this);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae)
    {
        try {
            int a = Integer.parseInt(tf1.getText());
            int b = Integer.parseInt(tf2.getText());
            if(b==0)
                throw new ArithmeticException(" Divide by Zero Error");
            float c = (float) a / b;
        }
    }
}
```

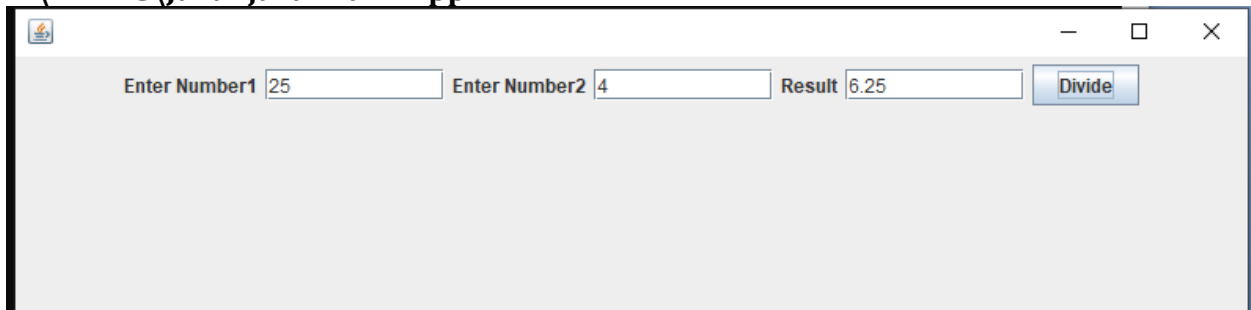
CS408PC: JAVA PROGRAMMING LAB

```
tf3.setText(String.valueOf(c));  
} catch (NumberFormatException ex)  
{  
    JOptionPane.showMessageDialog(this, ex.getMessage());  
} catch (ArithmeticException ex)  
{  
    JOptionPane.showMessageDialog(this, ex.getMessage());  
}  
}  
}  
public class IntDivApp  
{  
    public static void main(String[] args)  
    {  
        IntDiv a = new IntDiv();  
    }  
}
```

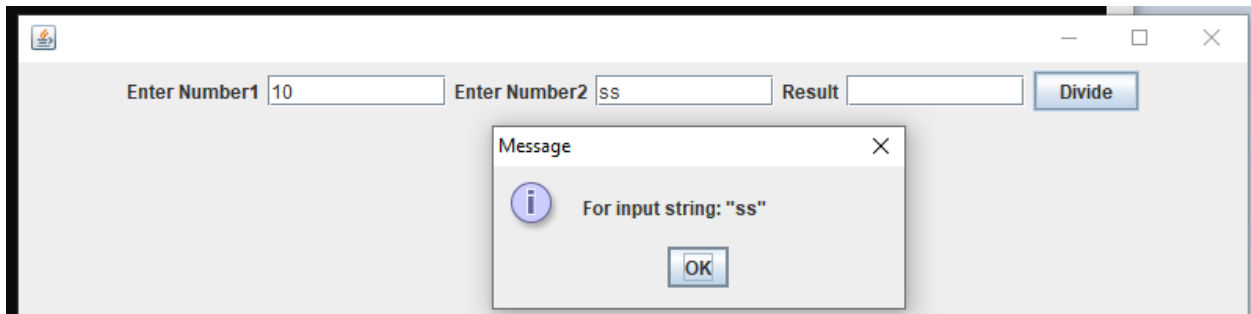
OUTPUT:

D:\TKREC\Java>javac IntDivApp.java

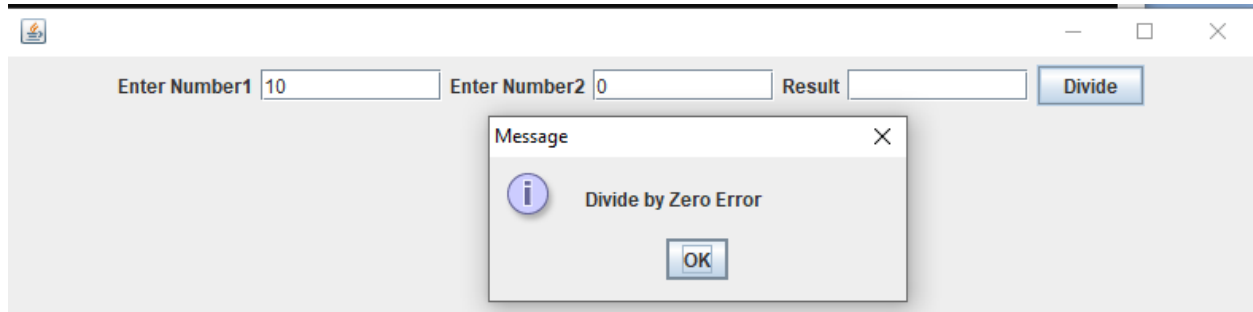
D:\TKREC\Java>java IntDivApp



Throwing Number Format Exception:



Throwing Arithmetic Exception:



5.1 OBJECTIVE:

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.

5.2 RESOURCES:

JDK1.8.0 is required, 1GB RAM, Hard Disk 80 GB

5.3 PROGRAM LOGIC:

1. Start the program
2. Create a class with the name „even implements Runnable“ and „odd implements Runnable“.
3. Create thread objects and Random class object.
4. Pass the objects of our class to thread class.

5.4 PROCEDURE:

To execute a java program we require setting a class path:

- 1.C:\set path= C:\Program Files\Java\jdk1.8.0\bin;.;
- 2.C:\javac Mthread.java
- 3.C:\java Mthread

5.5 SOURCE CODE:

```
import java.util.*;
```

```
class Even implements Runnable
{
    int x;
    Even(int x)
    {
        this.x = x;
    }
    public void run()
    {
```

CS408PC: JAVA PROGRAMMING LAB

```
System.out.println("Thread Name:Even Thread and " + x + " is even Number and Square of " +
x+ " is: " + x * x);
}
}
class Odd implements Runnable
{
    int x;
    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
{
    String tname;
    Random r;
    Thread t1, t2;

    A(String s)
    {
        tname = s;
    }
    public void run()
    {
        int num = 0;
        r = new Random();
        try
        {
            for (int i = 0; i < 5; i++)
            {
                num = r.nextInt(10);
                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(10);
    System.out.println("                ");
}
} catch (Exception ex)
{
    System.out.println(ex.getMessage());
}
}
}
public class Mthread
{
    public static void main(String[] args)
    {
        A a = new A("One");
        a.start();
    }
}
```

OUTPUT:

D:\TKREC\Java>javac Mthread.java

D:\TKREC\Java>java Mthread

Main Thread and Generated Number is 2

Thread Name:Even Thread and 2 is even Number and Square of 2 is: 4

Main Thread and Generated Number is 5

Thread Name:Odd Thread and 5 is odd number and Cube of 5 is: 125

Main Thread and Generated Number is 9

Thread Name:Odd Thread and 9 is odd number and Cube of 9 is: 729

Main Thread and Generated Number is 6

Thread Name:Even Thread and 6 is even Number and Square of 6 is: 36

Main Thread and Generated Number is 7

Thread Name:Odd Thread and 7 is odd number and Cube of 7 is: 343

Conclusion: Student will learn the concept of Multi-Threading.

Viva Question for Program 5

1. What is the difference between thread and process?
2. What are the techniques for multithreading?

3. What do you mean by Synchronization?
4. Explain Runnable interface.
5. What is the meaning of thread priority?

6.1 OBJECTIVE: Write a Java program for the following:

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

6.2 RESOURCES:

JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

6.3 PROGRAM LOGIC:

1. Start the program
2. Create a class with the name Node.
3. Create a class with the name LinkedList.
4. Create a class with the name DLLTest.
5. Create the Constructor for LinkedList
6. Call the LinkedList Constructor in main class i.e DLLTest

6.4 PROCEDURE:

To execute a java program we require setting a class path:

1. C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.;
2. C:\javac DLLTest .java
3. C:\java DLLTest

6.5 SOURCE CODE:

```
import java.util.Scanner;
class Node {
    protected int data;
    protected Node next, prev;

    Node() {
        next = null;
        prev = null;
        data = 0;
    }
    Node(int d, Node n, Node p) {
        data = d;
```

```
        next = n;
        prev = p;
    }
    public void setLinkNext(Node n) {
        next = n;
    }
    public void setLinkPrev(Node p) {
        prev = p;
    }
    public Node getLinkNext() {
        return next;
    }
    public Node getLinkPrev() {
        return prev;
    }
    public void setData(int d) {
        data = d;
    }
    public int getData() {
        return data;
    }
}

class LinkedList {
    protected Node start;
    protected Node end;
    public int size;

    LinkedList() {
        start = null;
        end = null;
        size = 0;
    }
    public boolean isEmpty() {
        return start == null;
    }
    public int getSize() {
        return size;
    }

    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++;
}

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++;
}

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++;
}

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();
}
}

public void display() {
    System.out.print("Doubly Linked List = ");
    if (size == 0) {
        System.out.print("empty");
        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");  
  }  
}  
public class DLLTest {  
    public static void main(String[] args) {  
        LinkedList list = new LinkedList();  
        System.out.println("Doubly Linked List Test");  
        System.out.println("\nInsert Element at Start");  
        list.insertAtStart(0);  
        list.display();  
        System.out.println("\nInsert at End");  
        list.insertAtEnd(5);  
        list.display();  
        System.out.println("\nInsert at Position");  
        list.insertAtPos(1, 2);  
        list.insertAtPos(2, 3);  
        list.insertAtPos(3, 4);  
        list.display();  
        System.out.println("\nDeleting at Position 1");  
        list.deleteAtPos(1);  
        list.display();  
    }  
}
```

OUTPUT:

D:\TKREC\Java>javac DLLTest.java

D:\TKREC\Java>java DLLTest

Doubly Linked List Test

Insert Element at Start

Doubly Linked List = 0

Insert at End

Doubly Linked List = 0 <-> 5

Insert at Position

Doubly Linked List = 0 <-> 1 <-> 2 <-> 3 <-> 5

Deleting at Position 1

Doubly Linked List = 1 <-> 2 <-> 3 <-> 5

CS408PC: JAVA PROGRAMMING LAB

7.1 OBJECTIVE

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.

7.2 RESOURCES:

JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

7.3 PROGRAM LOGIC:

1. Start the program
2. Create a class with the name A implements ItemListener.
3. Create the ButtonGroup ,JRadiobuttons,JPanel.
4. Add to the JFrame
5. Register the components to the JFrame.
6. Close the JFrame.

7.4 PROCEDURE:

To execute a java program we require setting a class path:

- 1.C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.;
- 2.C:\javac TLights.java
- 3.C:\java TLights

7.5 SOURCE CODE:

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

```
class A extends JFrame implements ItemListener
{
    JLabel l1, l2;
    JRadioButton r1, r2, r3;
    ButtonGroup bg;
    JPanel p, p1;

    A()
    {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(2, 1));
        setSize(800, 200);
        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);
```

CS408PC: JAVA PROGRAMMING LAB

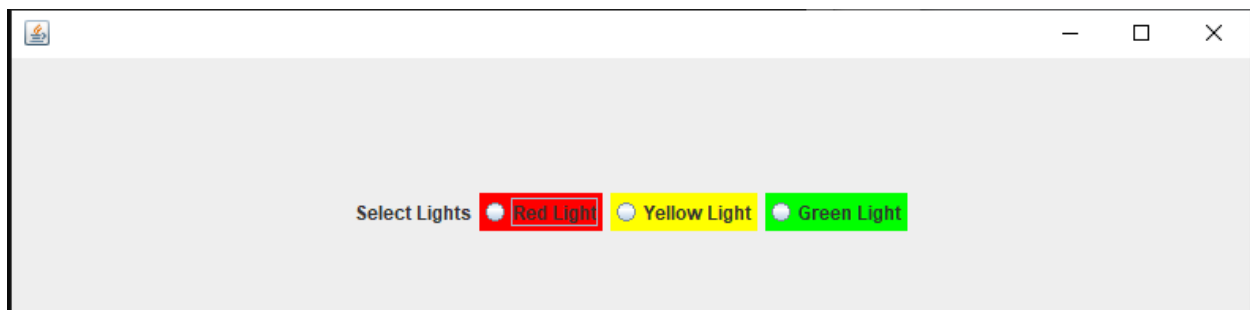
```
    }  
    break;  
    case "Yellow Light": {  
        l1.setText("Ready");  
        l1.setForeground(Color.YELLOW);  
    }  
    break;  
    case "Green Light": {  
        l1.setText("GO");  
        l1.setForeground(Color.GREEN);  
    }  
    break;  
    }  
}
```

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

OUTPUT:

D:\TKREC\Java>javac TLights.java

D:\TKREC\Java>java TLights



CS408PC: JAVA PROGRAMMING LAB



8.1 OBJECTIVE:

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.

8.2 RESOURCES:

JDK1.8.0 is required, 1GB RAM, Hard Disk 80 GB

8.3 PROGRAM LOGIC:

1. Start the program
2. Create a classes Shape, Rectangle, Triangle, Circle and AreaTest

8.4 PROCEDURE:

To execute a java program we require setting a class path:

1. C:\set path= C:\Program Files\Java\jdk1.8.0\bin;.;
2. C:\javac AreaTest.java
3. C:\java AreaTest

8.5 SOURCE CODE:

```
abstract class Shape
{
    int x, y;
    abstract void printArea();
}
class Rectangle extends Shape
{
    public void printArea()
    {
        System.out.println("\nArea of Rectangle is " + x * y);
    }
}
class Triangle extends Shape
{
    public void printArea()
    {
        System.out.println("\nArea of Triangle is " + (x * y) / 2);
    }
}
class Circle extends Shape
{
    public void printArea()
    {
        System.out.println("\nArea of Circle is " + (22 * x * x) / 7);
    }
}
public class AreaTest
{
    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:

D:\TKREC\Java>javac AreaTest.java

D:\TKREC\Java>java AreaTest

Area of Rectangle is 200

Area of Triangle is 525

Area of Circle is 12

9.1 OBJECTIVE

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.

9.2 RESOURCES:

JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

9.3 PROGRAM LOGIC:

1. Start the program
2. Create a class with the name Aa extends JFrame.

9.4 PROCEDURE:

To execute a java program we require setting a class path:

1. C:\set path= C:\Program Files\Java\jdk1.6.0\bin.;
2. C:\javac Test.java
3. C:\java Test

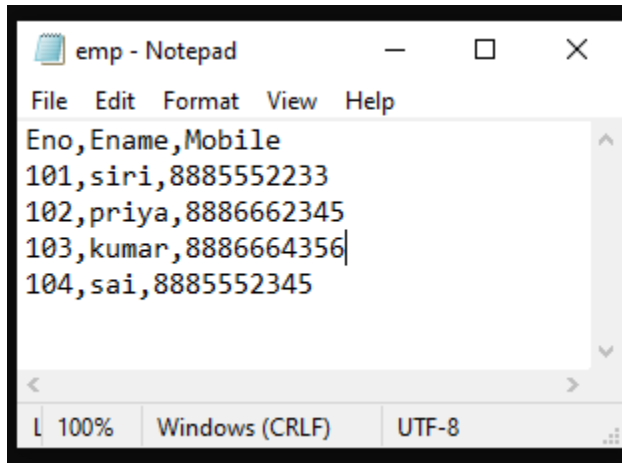
9.5 SOURCE CODE:

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

class Aa extends JFrame
{
    Aa()
    {
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        GridLayout g = new GridLayout(0, 3);
        setLayout(g);
        try{
            FileInputStream in = new FileInputStream("D:\\emp.txt");
            Scanner sc = new Scanner(in).useDelimiter(",");
            String[] arrayList;
            String a;
            while (sc.hasNextLine()) {
                a = sc.nextLine();
                arrayList = a.split(",");
                for (String i : arrayList) {
                    add(new JLabel(i));
                }
            }
        } catch (Exception e) {
        }
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
}

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

Emp.txt



```
emp - Notepad
File Edit Format View Help
Eno,Ename,Mobile
101,siri,8885552233
102,priya,8886662345
103,kumar,8886664356
104,sai,8885552345
100% Windows (CRLF) UTF-8
```

OUTPUT:

D:\>javac Test.java

D:\>java Test



```
D:\TKREC\Java>cd..
D:\TKREC>cd..
D:\>javac Test.java
D:\>java Test
```

Eno	Ename	Mobile
101	siri	8885552233
102	priya	8886662345
103	kumar	8886664356
104	sai	8885552345

10.1 OBJECTIVE:

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)

10.2 RESOURCES:

JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

10.3 PROGRAM LOGIC:

1. Start the program
2. Create a class with the name A implements MouseListener.
3. Call the mouse events.
4. Add to the JFrame
5. Register the components to the JFrame.

6. Close the JFrame.

10.4 PROCEDURE:

To execute a java program we require setting a class path:

- 1.C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.;
- 2.C:\javac Test.java
- 3.C:\java Test

10.5 SOURCE CODE:

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

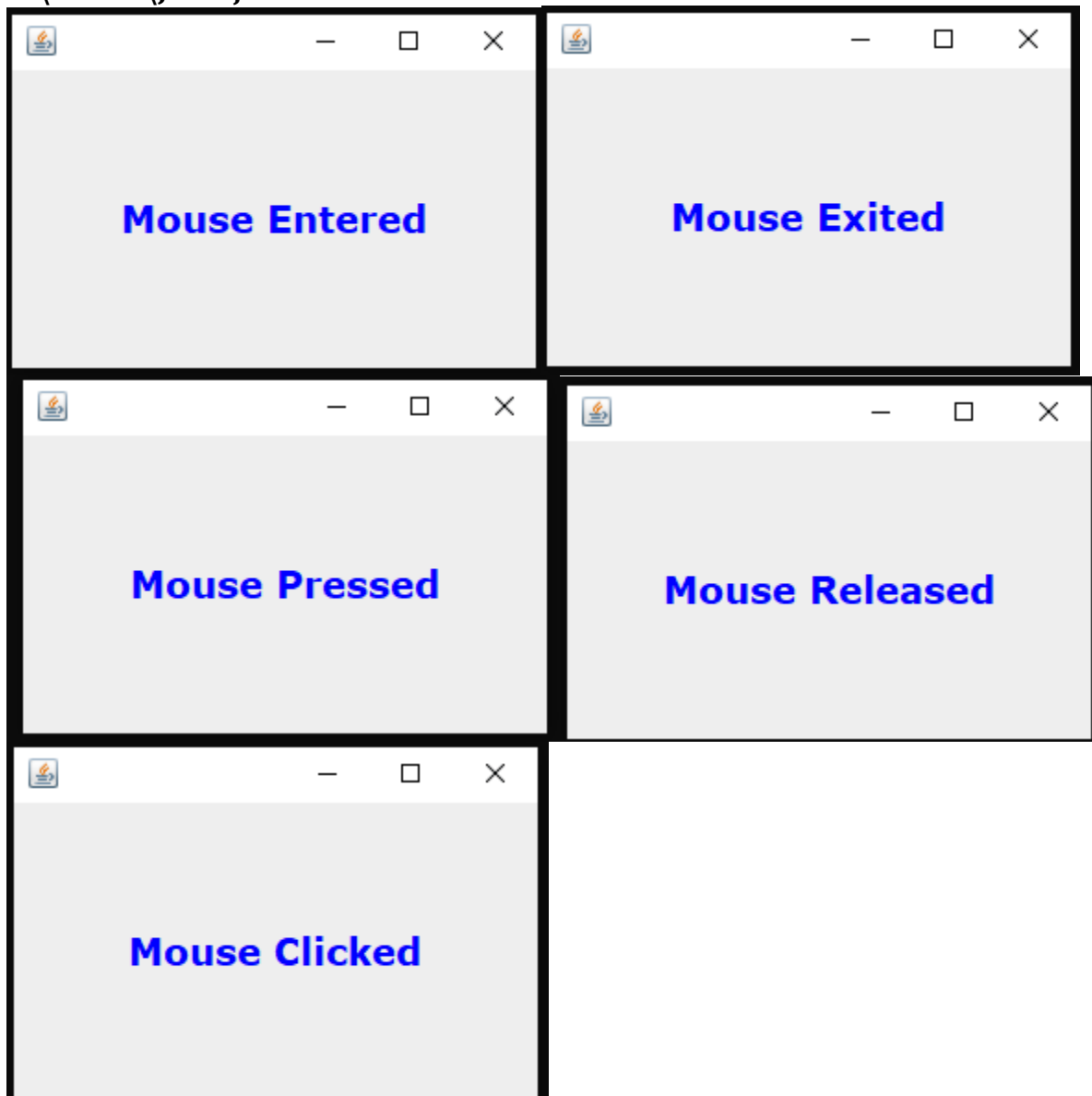
class A extends JFrame implements MouseListener
{
    JLabel l1;
    A() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 200);
        setLayout(new GridBagLayout());
        l1 = new JLabel();
        Font f = new Font("Verdana", Font.BOLD, 20);
        l1.setFont(f);
        l1.setForeground(Color.BLUE);
        l1.setAlignmentX(Component.CENTER_ALIGNMENT);
        l1.setAlignmentY(Component.CENTER_ALIGNMENT);
        add(l1);
        addMouseListener(this);
        setVisible(true);
    }
    public void mouseExited(MouseEvent m) {
        l1.setText("Mouse Exited");
    }
    public void mouseEntered(MouseEvent m) {
        l1.setText("Mouse Entered");
    }
    public void mouseReleased(MouseEvent m) {
        l1.setText("Mouse Released");
    }
    public void mousePressed(MouseEvent m) {
        l1.setText("Mouse Pressed");
    }
    public void mouseClicked(MouseEvent m) {
        l1.setText("Mouse Clicked");
    }
}
```

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

OUTPUT:

D:\TKREC\Java>javac javac Test.java

D:\TKREC\Java>java Test



11.1 OBJECTIVE:-

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).

11.2 RESOURCES:

JDK1.7.0 is required, 1GB RAM, Hard Disk 80 GB

11.3 PROGRAM LOGIC:

1. Start the program
2. Create a class with the name Hashtbl
3. Create FileInputStream, Scanner, Hashtable objects
4. Read the data from the hash table.
5. Display the data.

11.4 PROCEDURE:

To execute a java program we require setting a class path:

- 1.C:\set path= C:\Program Files\Java\jdk1.6.0\bin;;
- 2.C:\javac Hashtbl.java
- 3.C:\java Hashtbl

11.5 SOURCE CODE:

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

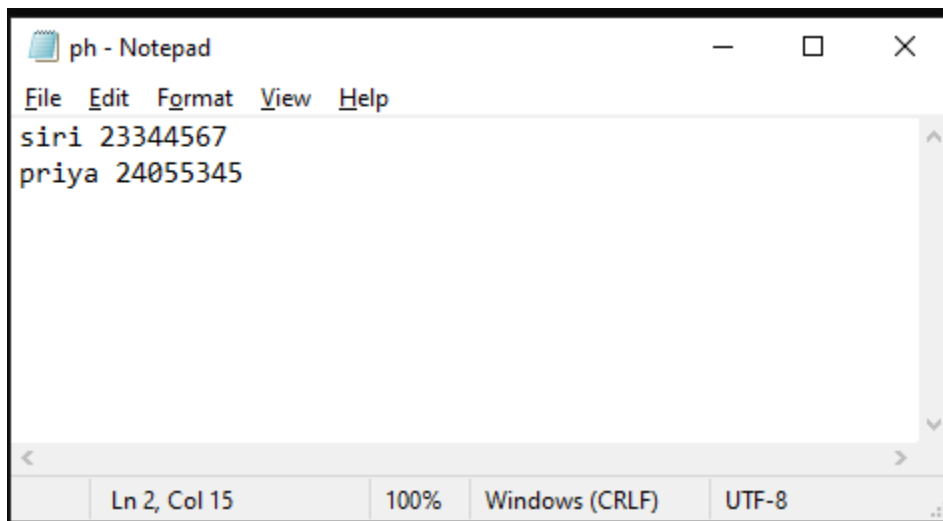
public class Hashtbl
{
    public static void main(String[] args)
    {
        try {
            FileInputStream in = new FileInputStream("D:\\ph.txt");
            Scanner sc = new Scanner(in).useDelimiter("\\s+");
            Hashtable<String, String> ht = new Hashtable<String, String>();
            String[] arrayList;
            String a;
            System.out.println("Welcome To Java Programming");
            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 Java Programming");
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:

D:\Siri>javac Hashtbl.java

D:\Siri>java Hashtbl

Welcome To Java Programming

HASH TABLE IS

KEY : VALUE

siri:23344567

priya:24055345

Welcome TO Java Programming

----MENU-----

----1.Search by Name-----

----2.Search by Mobile-----

----3.Exit-----

Enter Your Option 1,2,3

1

Enter Name

siri

Mobile is 23344567

Enter Your Option 1,2,3

24055345

Choose Option between 1 and Three

Enter Your Option 1,2,3

2

Enter mobile

24055345

Name is priya

Enter Your Option 1,2,3

3

Menu Successfully Exited

12.1 OBJECTIVE:

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

12.2 RESOURCES:

JDK1.8.0 is required, 1GB RAM, Hard Disk 80 GB

12.3 PROGRAM LOGIC:

1. Start the program
2. Create a class with the name producer implements Runnable and consumer implements Runnable.
3. Create ItemQueue class object.
4. Pass the object to thread classes.

12.4 PROCEDURE:

To execute a java program we require setting a class path:

- 1.C:\set path= C:\Program Files\Java\jdk1.8.0\bin,; ;
- 2.C:\javac ProducerConsumer.java
- 3.C:\java ProducerConsumer

12.5 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:

D:\TKREC\Java>javac ProducerConsumer.java

D:\TKREC\Java>java ProducerConsumer

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

And so on continuation, if we interrupt then it will stopped

13.1 OBJECTIVE:

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

13.2 RESOURCES:

JDK1.8.0 is required, 1GB RAM, Hard Disk 80 GB

13.3 PROGRAM LOGIC:

1. Start the program
2. Create a class with the name ListingFiles
3. Create Scanner class object.

4. Read the files from the Directory and subdirectory.

13.4 PROCEDURE:

To execute a java program we require setting a class path:

- 1.C:\set path= C:\Program Files\Java\jdk1.8.0\bin;.;
- 2.C:\javac ListingFiles.java
- 3.C:\java ListingFiles

13.5 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:

D:\TKREC\Java>javac ListingFiles.java

D:\TKREC\Java>java ListingFiles

Enter the root directory name: d

Current Location: d:\

- !qhlogs.doc

> \$RECYCLE.BIN

- Aa.class

- AccountBalance.java

> B Tech Material

> Big Data

> com

- emp.txt

- file.txt

> GK

CS408PC: JAVA PROGRAMMING LAB

- > Greenfoot games
- Hashtbl.class
- Hashtbl.java
- ITR3_2018.zip
- > Java
- > Java Eclipse
- > NAAC
- > NBA
- > NBA-2019
- NBA.rar
- ph.txt
- > Phd
- > Priya
- > SDES
- > SIRI
- > Softwares
- > System Volume Information
- > TASK
- Test.class
- Test.java
- > TKREC
- > Vishwas
- Windows 7 key.txt

Do you want to open any sub-directory (Y/N): y

Enter the sub-directory name: tkrec

Current Location: d:\\\\tkrec

- 4-2 TIME TABLE.pdf
- 492Academic calendar 2021.pdf
- Bloom's Pic(print).docx
- Blooms Tax.pdf
- format.xlsx
- > I Year
- > II Year
- > III Year
- INSTRUCTIONS TO STUDENTS FOR APPEARING ONLINE EXAMS-MAY 2021.docx
- > IV Year
- IV-II MID 1 QP FORMAT.docx
- IV-II MID INSTRUCTIONS.docx
- > Java
- JTNUH BTech B.Pharmacy AC 2021 Revised.pdf
- > LIC
- Links for study.docx
- > MBA
- > NBA
- PayUMoney Payment Page-exam fee.pdf

CS408PC: JAVA PROGRAMMING LAB

- QualityRequirementsandExpectationsofSoftwareDevelopment.pdf

- subject distribution.xlsx

- ~\$oom's Pic(print).docx

Do you want to open any sub-directory (Y/N): y

Enter the sub-directory name: II Year

The sub-directory does not exists!

Current Location: d:\\\\tkrec\\

- 4-2 TIME TABLE.pdf

- 492Academic calendar 2021.pdf

- Bloom's Pic(print).docx

- Blooms Tax.pdf

- format.xlsx

> I Year

> II Year

> III Year

- INSTRUCTIONS TO STUDENTS FOR APPEARING ONLINE EXAMS-MAY 2021.docx

> IV Year

- IV-II MID 1 QP FORMAT.docx

- IV-II MID INSTRUCTIONS.docx

> Java

- JTNUH BTech B.Pharmacy AC 2021 Revised.pdf

> LIC

> MBA

> NBA

- PayUMoney Payment Page-exam fee.pdf

- QualityRequirementsandExpectationsofSoftwareDevelopment.pdf

- subject distribution.xlsx

- ~\$oom's Pic(print).docx

Do you want to open any sub-directory (Y/N): ***** Program Closed *****

14.1 OBJECTIVE:

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

14.2 RESOURCES:

JDK1.8.0 is required, 1GB RAM, Hard Disk 80 GB

14.3 PROGRAM LOGIC:

1. Start the program
2. Create a class with the name QuickSortOnStrings
3. Create QuickSortOnStrings class object.
4. Initialize the list of strings names and sort these names

14.4 PROCEDURE:

To execute a java program we require setting a class path:

- 1.C:\set path= C:\Program Files\Java\jdk1.8.0\bin,.;
- 2.C:\javac QuickSortOnStrings.java
- 3.C:\java QuickSortOnStrings

14.5 SOURCE CODE:

```
public class QuickSortOnStrings
{
    String names[];
    int length;

    public static void main(String[] args)
    {
        QuickSortOnStrings obj = new QuickSortOnStrings();
        String stringsList[] = {"siri", "gouthu", "rani", "gouthami", "honey", "priya", "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:

D:\TKREC\Java>javac QuickSortOnStrings.java

D:\TKREC\Java>java QuickSortOnStrings
gouthami gouthu hello honey priya rani siri

15.1 OBJECTIVE:

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

15.2

JDK1.8.0 is required, 1GB RAM, Hard Disk 80 GB

15.3 PROGRAM LOGIC:

1. Start the program
2. Create a class with the name BubbleSort
3. Create Scanner class object.

4. Read the values from console and sort it the elements.

15.4 PROCEDURE:

To execute a java program we require setting a class path:

- 1.C:\set path= C:\Program Files\Java\jdk1.8.0\bin;.;
- 2.C:\javac BubbleSort.java
- 3.C:\java BubbleSort

15.5 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++;
                }
            }
        }
    }
}
```

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

OUTPUT:

D:\TKREC\Java>javac BubbleSort.java

D:\TKREC\Java>java BubbleSort

Enter the list size: 6

Enter any 6 integer numbers:

1 5 3 4 9 6

List of sorted elements:

9 6 5 4 3 1

Total number of Interchanges is 12

