**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.**

**//Program to print even and odd numbers in first 10 natural numbers**

public class Test {

public static void main(String[] args) {

int num = 10;

for(int i = 1;i<=10;i++){

if(i%2==0){

System.out.println(i+" even");

}

else{

System.out.println(i+" odd");

}

}

}

}

**Output**:

1 odd

2 even

3 odd

4 even

5 odd

6 even

7 odd

8 even

9 odd

10 even

**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 theresult. Handle any possible**

**exceptions like divided by zero.**

**Program:**

Package TestProject; import javax.swing.JFrame;

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

class BuildCalculator extends JFrame implements ActionListener{ JFrame actualWindow;

JPanel resultPanel, buttonPanel, infoPanel;

JTextField resultTxt;

JButton btn\_digits[] = new JButton[10];

JButton btn\_plus, btn\_minus, btn\_mul, btn\_div, btn\_equal, btn\_dot, btn\_clear;

char eventFrom;

JLabel expression, appTitle, siteTitle ;

double oparand\_1 = 0, operand\_2 = 0; String operator = "=";

BuildCalculator() {

Font txtFont = new Font("TimesNewroman", Font.***BOLD***, 20); Font titleFont = new Font("", Font.***BOLD***, 30);

Font expressionFont = new Font("", Font.***BOLD***, 15);

actualWindow = new JFrame("Calculator"); resultPanel = new JPanel();

buttonPanel = new JPanel(); infoPanel = new JPanel();

actualWindow.setLayout(new GridLayout(3, 1));

buttonPanel.setLayout(new GridLayout(4, 4));

infoPanel.setLayout(new GridLayout(3, 1)); actualWindow.setResizable(false);

appTitle = new JLabel("Basic Calculator"); appTitle.setFont(titleFont);

expression = new JLabel("Expression shown here"); expression.setFont(expressionFont);

siteTitle = new JLabel("SimpleCalculator"); siteTitle.setFont(expressionFont); siteTitle.setHorizontalAlignment(SwingConstants.***CENTER***); siteTitle.setForeground(Color.***BLUE***);

resultTxt = new JTextField(15); resultTxt.setBorder(null); resultTxt.setPreferredSize(new Dimension(15, 50));

resultTxt.setFont(txtFont);

resultTxt.setHorizontalAlignment(SwingConstants.***RIGHT***);

for(int i = 0; i < 10; i++) { btn\_digits[i] = new JButton(""+i);

btn\_digits[i].addActionListener(this);

}

btn\_plus = new JButton("+");

btn\_plus.addActionListener(this); btn\_minus = new JButton("-"); btn\_minus.addActionListener(this); btn\_mul = new JButton("\*"); btn\_mul.addActionListener(this); btn\_div = new JButton("/"); btn\_div.addActionListener(this); btn\_dot = new JButton("."); btn\_dot.addActionListener(this); btn\_equal = new JButton("="); btn\_equal.addActionListener(this); btn\_clear = new JButton("Clear"); btn\_clear.addActionListener(this);

resultPanel.add(appTitle); resultPanel.add(resultTxt); resultPanel.add(expression); for(int i = 0; i < 10; i++) {

buttonPanel.add(btn\_digits[i]);}

buttonPanel.add(btn\_plus); buttonPanel.add(btn\_minus);

buttonPanel.add(btn\_mul); buttonPanel.add(btn\_div); buttonPanel.add(btn\_dot); buttonPanel.add(btn\_equal); infoPanel.add(btn\_clear); infoPanel.add(siteTitle);

actualWindow.add(resultPanel); actualWindow.add(buttonPanel); actualWindow.add(infoPanel);

actualWindow.setSize(300, 500); actualWindow.setVisible(true);

}

*@Override*

public void actionPerformed(ActionEvent e) { eventFrom = e.getActionCommand().charAt(0); String buildNumber;

if(Character.*isDigit*(eventFrom))

buildNumber = resultTxt.getText() + eventFrom; resultTxt.setText(buildNumber);

} else if(e.getActionCommand() == ".") { buildNumber = resultTxt.getText() + eventFrom; resultTxt.setText(buildNumber);

}

else if(eventFrom != '='){

oparand\_1 = Double.*parseDouble*(resultTxt.getText()); operator = e.getActionCommand(); expression.setText(oparand\_1 + " " + operator); resultTxt.setText("");

} else if(e.getActionCommand() == "Clear") { resultTxt.setText("");

}

else {

operand\_2 = Double.*parseDouble*(resultTxt.getText());

expression.setText(expression.getText() + " " + operand\_2); switch(operator) {

case "+": resultTxt.setText(""+(oparand\_1 + operand\_2));

break;

case "-": resultTxt.setText(""+(oparand\_1 - operand\_2));

break;

case "\*": resultTxt.setText(""+(oparand\_1 \* operand\_2));

break;

case "/": try { if(operand\_2 == 0)

throw new ArithmeticException(); resultTxt.setText(""+(oparand\_1 / operand\_2)); break;

} catch(ArithmeticException ae) {JOptionPane.*showMessageDialog*(actualWindow, "Divisor can

not be ZERO");

} }

}

}

}

public class Calculator {

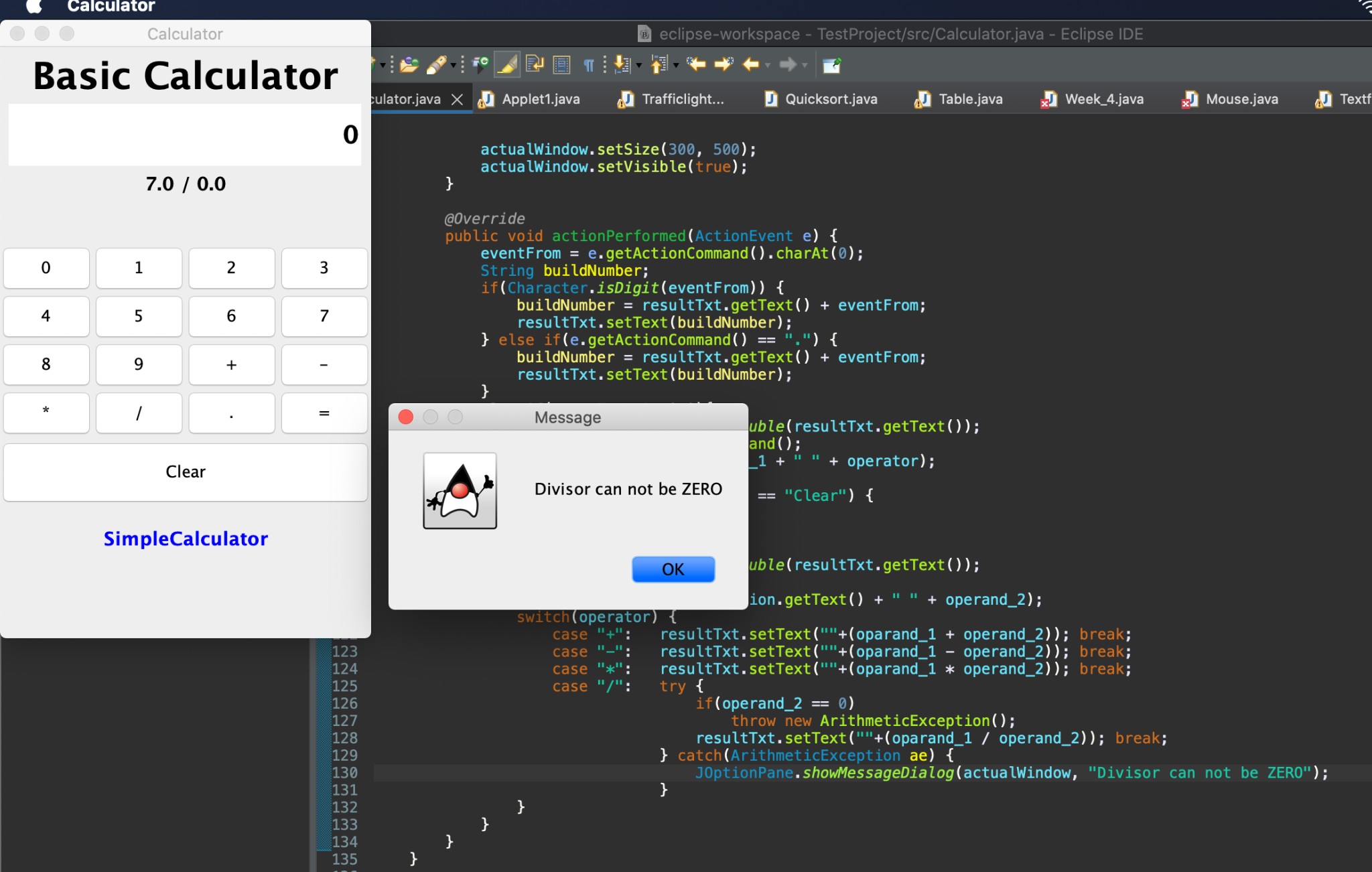
public static void main(String[] args) {

new BuildCalculator();

}

}

**Output:**



**3.Develop an applet in Java that displays a simple message.**

**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.**

import java.awt.\*;

import java.applet.\*;

/\*

<applet code="Demo" width=300 height=200>

</applet>

\*/

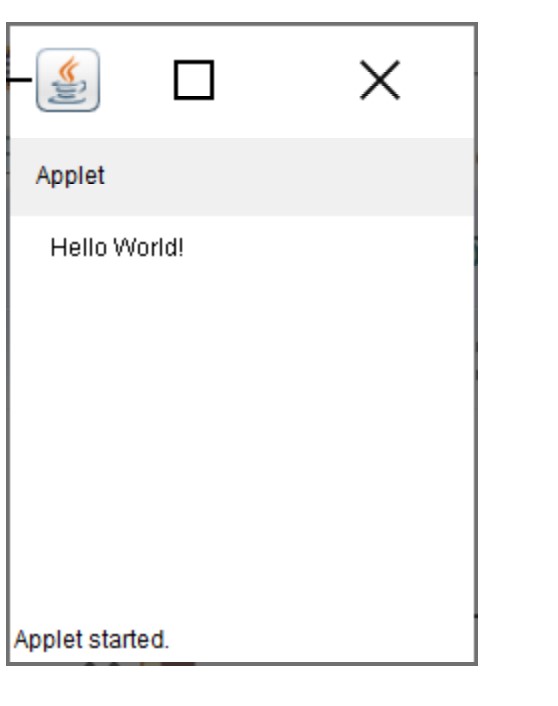
public class Applet1 extends Applet {

public void paint(Graphics g) {

g.drawString("Welcome to Goeduhub Technologies", 20, 20);

}

}

**Output**:

**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.**

import java.awt.\*;

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

class BuildGUI extends JFrame implements ActionListener {

JFrame actualWindow;

JPanel container;

JTextField txt\_num1, txt\_num2, txt\_result;

JButton btn\_div;

BuildGUI() { actualWindow = new JFrame("Experiment 4"); container = new JPanel(); container.setLayout(new FlowLayout());

txt\_num1 = new JTextField(20);

txt\_num2 = new JTextField(20); txt\_result = new JTextField(20);

btn\_div = new JButton("Divide"); btn\_div.addActionListener(this);

container.add(txt\_num1); container.add(txt\_num2); container.add(btn\_div); container.add(txt\_result);

actualWindow.add(container); actualWindow.setSize(300, 300); actualWindow.setVisible(true);

}

*@Override* public void actionPerformed(ActionEvent e) { int num1,num2;

try {

num1 = Integer.*parseInt*(txt\_num1.getText()); num2 = Integer.*parseInt*(txt\_num2.getText()); txt\_result.setText(num1/num2+"");}

catch(NumberFormatException nfe) {

JOptionPane.*showMessageDialog*(actualWindow,"Please do enter only

integers");

}

catch(ArithmeticException ae) {

JOptionPane.*showMessageDialog*(actualWindow,"Divisor can not be ZERO");

}

}

}

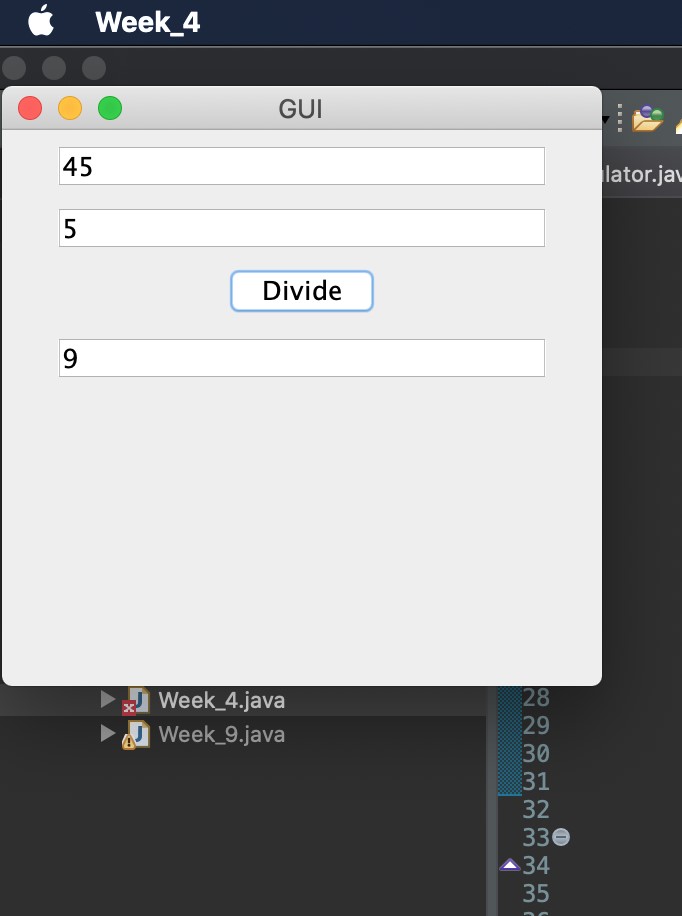
public class Week\_4 {

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

}

**}**

**Output:**

****

**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.**

import java.util.Random;

class Square extends Thread{

int num;

Square(int num){

this.num = num; }

public void run() {

System.out.println("Square of "+num+"is "+num\*num);

System.out.println(); }

}

class Cube extends Thread{

int num;

Cube(int num){

this.num = num; }

public void run() {

System.out.println("Cube of "+num+" is "+num\*num\*num);

System.out.println(); }

}

class Rand implements Runnable{

public void run() {

Random r = new Random();

for(int i = 0;i<10;i++) {

int ranint = r.nextInt(1000);

System.out.println("Number generated is "+ranint);

if(ranint%2 == 0) {

Square s = new Square(ranint);

s.run(); }

else {

Cube c = new Cube(ranint);

c.run(); }

try {

Thread.sleep(1000); }

catch(Exception e) {

System.out.println(e); }

}

}

}

public class Test {

public static void main(String[] args) {

Rand rn = new Rand();

rn.run();

}

}

**Output:**

Number generated is 17

Cube of 17 is 4913

Number generated is 505

Cube of 505 is 128787625

Number generated is 236

Square of 236is 55696

Number generated is 45

Cube of 45 is 91125

Number generated is 864

Square of 864is 746496

Number generated is 236

Square of 236is 55696

Number generated is 876

Square of 876is 767376

Number generated is 235

Cube of 235 is 12977875

**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.**

**Program:**

public class DoubleLinkedList {

class Node {

int data;

Node previous;

Node next;

public Node(int data) {

this.data = data;

}

}

Node head, tail = null;

public void addNode(int data) {

Node newNode = new Node(data);

if (head == null) {

head = tail = newNode;

head.previous = null;

tail.next = null;

} else {

tail.next = newNode;

newNode.previous = tail;

tail = newNode;

tail.next = null;

}

}

public void display() {

Node current = head;

if (head == null) {

System.out.println("List is empty");

return;

}

System.out.println("Nodes of doubly linked list: ");

while (current != null) {

System.out.print(current.data + " ");

current = current.next;

}

}

public static void main(String[] args) {

DoubleLinkedList dList = new DoubleLinkedList();

dList.addNode(1);

dList.addNode(2);

dList.addNode(3);

dList.addNode(4);

dList.addNode(5);

dList.display();

}

}

**Output:**

Nodes of doubly linked list:

1 2 3 4 5

**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.**

**Program:**

import java.awt.Color;

import java.awt.\*;

import java.awt.event.ItemEvent; import java.awt.event.ItemListener;

import javax.swing.\*;

class App extends JFrame implements ItemListener{ JFrame actualWindow;

JPanel messageContainer, lightsContainer; JLabel message;

ButtonGroup btn\_group;

JRadioButton rb\_red, rb\_yellow, rb\_green;

App() {

Font myFont = new Font("Verdana",Font.BOLD, 30);

actualWindow = new JFrame("Traffic Lights");

messageContainer = new JPanel();

lightsContainer = new JPanel();

message = new JLabel("Select Light"); btn\_group = new ButtonGroup(); rb\_red = new JRadioButton("Red");

rb\_yellow = new JRadioButton("Yellow"); rb\_green = new JRadioButton("Green");

actualWindow.setLayout(new GridLayout(2, 1));

message.setFont(myFont); rb\_red.setForeground(Color.RED); rb\_yellow.setForeground(Color.YELLOW); rb\_green.setForeground(Color.GREEN);

btn\_group.add(rb\_red); btn\_group.add(rb\_yellow); btn\_group.add(rb\_green);

rb\_red.addItemListener(this); rb\_yellow.addItemListener(this);

b\_green.addItemListener(this);

messageContainer.add(message); lightsContainer.add(rb\_red); lightsContainer.add(rb\_yellow); lightsContainer.add(rb\_green);

actualWindow.add(messageContainer); actualWindow.add(lightsContainer);

actualWindow.setSize(300, 200); actualWindow.setVisible(true);

}

@Override

public void itemStateChanged(ItemEvent ie) {

JRadioButton selected = (JRadioButton) ie.getSource(); String textOnButton = selected.getText(); if(textOnButton.equals("Red")) {

message.setForeground(Color.RED); message.setText("STOP");

} else if(textOnButton.equals("Yellow")) {

message.setForeground(Color.YELLOW); message.setText("READY");

} else {

message.setForeground(Color.GREEN); message.setText("GO");

}

}

}

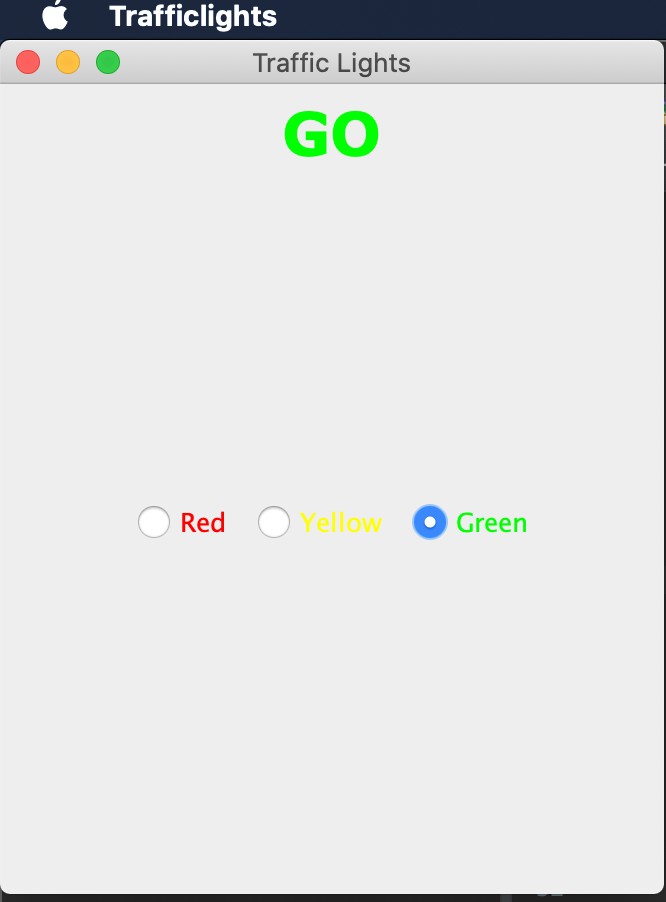
public class Trafficlights {

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

}

}

**Output:**



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

import java.util.Scanner;

import java.io.\*;

abstract class Shape{

public int a;

public int b;

abstract void printArea(int a, int b); }

class Rectangle extends Shape{

void printArea(int a,int b){

System.out.println("Area of rectangle of length "+a+" breadth "+b+" is "+(a\*b)+" units"); } }

class Triangle extends Shape{

void printArea(int a,int b){

System.out.println("Area of triangle of height "+a+" base "+b+" is "+(0.5\*a\*b)+" units"); } }

class Circle extends Shape{

void printArea(int a,int b){

System.out.println("Area of circle of radius "+a+" is "+(3.14\*a\*a)+" units"); } }

public class Main{

public static void main(String[] args){

Rectangle r = new Rectangle();

Triangle t = new Triangle();

Circle c = new Circle();

r.printArea(4,3);

t.printArea(7,5);

c.printArea(4,0);

} }

**Output:**

Area of rectangle of length 4 breadth 3 is 12 units

Area of triangle of height 7 base 5 is 17.5 units

Area of circle of radius 4 is 50.24 units

**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.**

**Progream:**

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

class A extends JFrame { public A() {

setSize(400, 400); setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***); GridLayout g = new GridLayout(0, 3);

setLayout(g);

try {

FileInputStream fin = new

FileInputStream("C:\\Users\\User\\eclipse-workspace\\HashTab.txt"); Scanner sc = new Scanner(fin).useDelimiter(",");

String[] arrayList;

String a;

while (sc.hasNextLine()) {

a = sc.nextLine(); arrayList = a.split(",");

for (String i : arrayList) {

add(new JLabel(i)); }

}

} catch (Exception ex) {

}

*setDefaultLookAndFeelDecorated*(true); pack();

setVisible(true);

}

}

public class Week\_9 {

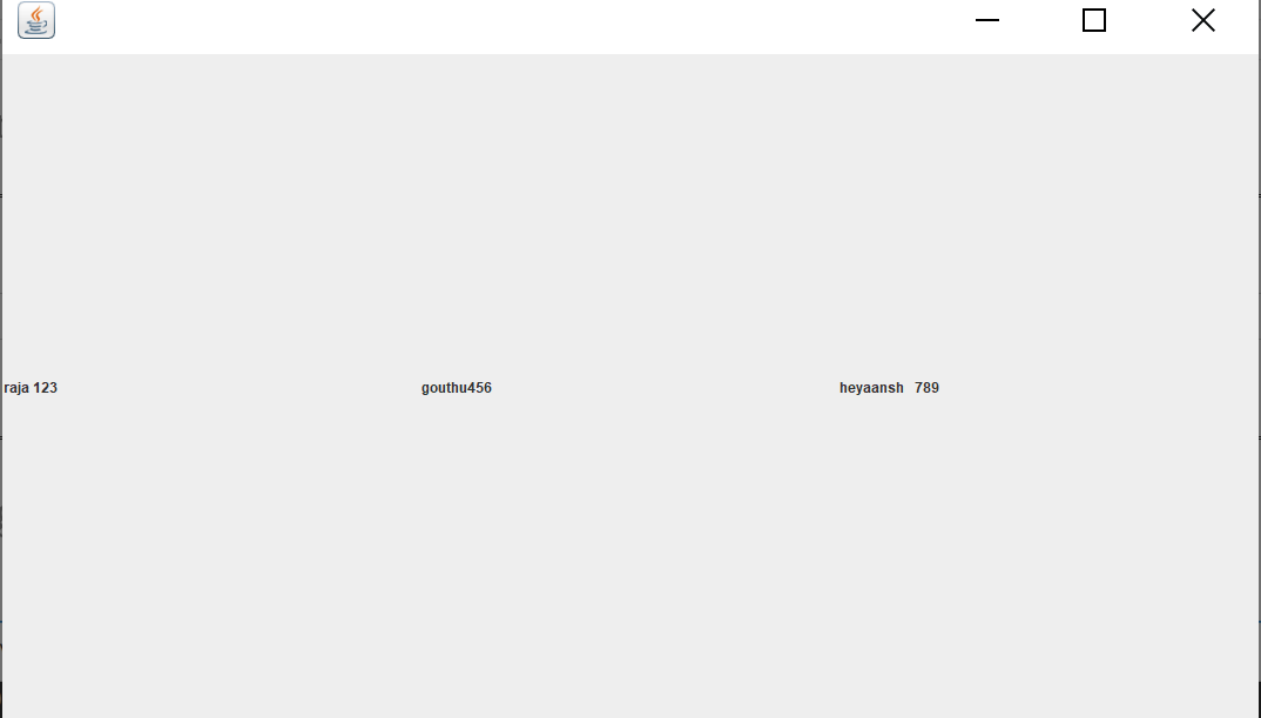
public static void main(String[] args) {

A a = new A();

}

}

**Output::**



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

**Program:**

import java.awt.Font;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import javax.swing.\*;

class App extends JFrame implements MouseListener { JFrame actualWindow;

JLabel message;

App() {

Font myFont = new Font("Verdana",Font.***BOLD***, 30); actualWindow = new JFrame("Mouse Tracking"); message = new JLabel("Mouse Events");

actualWindow.addMouseListener(this);

message.setFont(myFont); message.setHorizontalAlignment(JLabel.***CENTER***);

actualWindow.add(message);

actualWindow.setSize(500, 500); actualWindow.setVisible(true); }

*@Override*

public void mouseClicked(MouseEvent arg0) { message.setText("Mouse Clicked"); }

*@Override*

public void mouseEntered(MouseEvent arg0) { message.setText("Mouse Entered"); }

*@Override*

public void mouseExited(MouseEvent arg0) { message.setText("Mouse Exited"); }

*@Override*

public void mousePressed(MouseEvent arg0) { message.setText("Mouse Pressed");

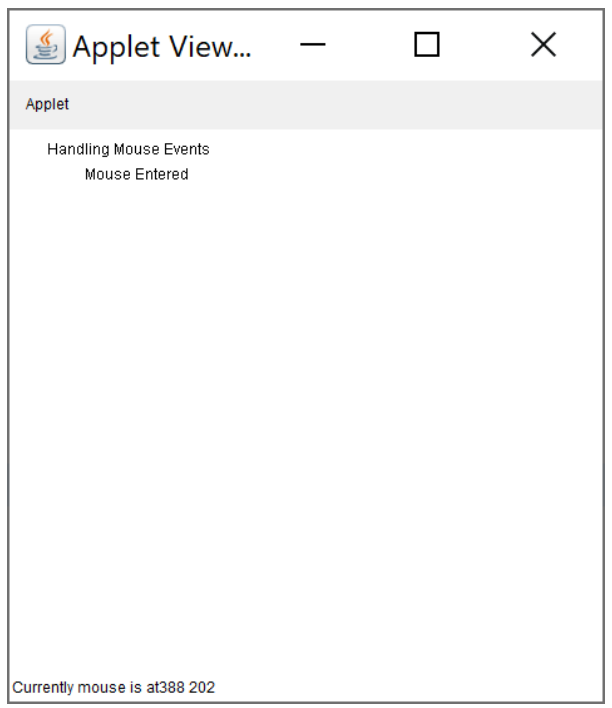
}*@Override*

public void mouseReleased(MouseEvent arg0) { message.setText("Mouse Released"); } }

public class Mouse{

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

} }

****

**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).**

Importjava.io.BufferedReader; import java.io.File;

import java.io.FileNotFoundException; import java.io.FileReader;

import java.io.IOException; import java.util.Hashtable; import java.util.Iterator; import java.util.Set;

public class Table{

public static void main(String[] args) { Table prog11 = new Table();

Hashtable<String, String> hashData = prog11.readFromFile("HashTab.txt"); System.***out***.println("File data into Hashtable:\n" + hashData); prog11.printTheData(hashData, "raja");

prog11.printTheData(hashData, "123"); prog11.printTheData(hashData, " ");

}

private void printTheData(Hashtable<String, String> hashData, String input) { String output = null;

if (hashData != null) {

Set<String> keys = hashData.keySet(); if (keys.contains(input)) {

output = hashData.get(input);

} else {

Iterator<String> iterator = keys.iterator(); while (iterator.hasNext()) {

String key = iterator.next();

String value = hashData.get(key); if (value.equals(input)) {

output = key; break;

System.***out***.println("Input given:" +input);

if (output != null) {

System.***out***.println("Data found in HashTable:" + output);

} else {

System.***out***.println("Data not found in HashTable");

}

}

private Hashtable<String, String> readFromFile(String fileName) { Hashtable<String, String> hashData = new Hashtable<String, String>(); try {

File f = new File("D:\\java\\" + fileName);

BufferedReader br = new BufferedReader(new FileReader(f)); String line = null;

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

String[] details = line.split("\t");

hashData.put(details[0], details[1]);

}

} catch (FileNotFoundException e) { e.printStackTrace();

} catch (IOException e) { e.printStackTrace();

}

return hashData;

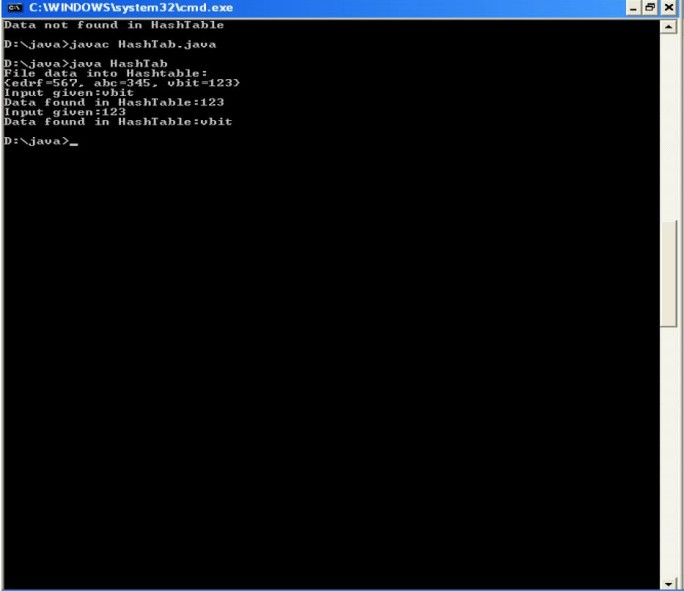
}

}

}

}

**Output:**



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

**Program:**

package multithreading;

import java.util.LinkedList;

public class ProducerAndConsumer {

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

final PC pc = new PC();

Thread t1 = new Thread(new Runnable() {

public void run() {

try {

pc.produce();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

});

Thread t2 = new Thread(new Runnable() {

public void run() {

try {

pc.consume();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

});

t1.start();

t2.start();

t1.join();

t2.join();

}

public static class PC {

LinkedList<Integer> list = new LinkedList<Integer>();

int capacity = 2;

public void produce() throws InterruptedException {

int value = 0;

while (true) {

synchronized (this) {

while (list.size() == capacity)

wait();

System.*out*.println("Producer produced-" + value);

list.add(value++);

notify();

Thread.*sleep*(1000);

}

}

}

public void consume() throws InterruptedException {

while (true) {

synchronized (this) {

while (list.size() == 0)

wait();

int val = list.removeFirst();

System.*out*.println("Consumer consumed-" + val);

notify();

Thread.*sleep*(1000);

}

}

}

}

}

**Output:**

Producer produced-0

Consumer consumed-0

Producer produced-1

Producer produced-2

Consumer consumed-1

Consumer consumed-2

Producer produced-3

Producer produced-4

Consumer consumed-3

Consumer consumed-4

Producer produced-5

Producer produced-6

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

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

public class Textfile {

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());

}

}

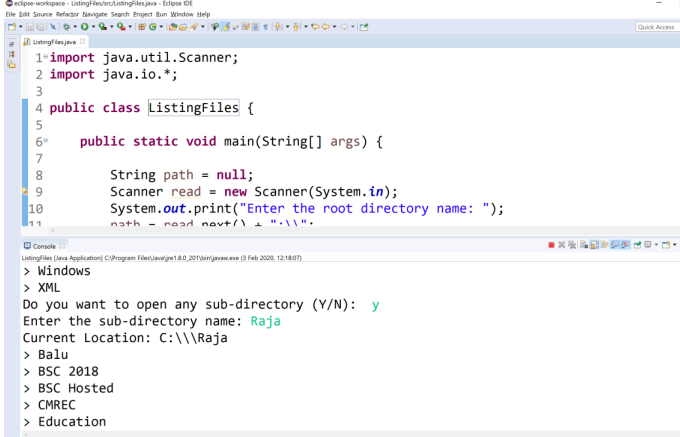
public static void printLine() {

System.***out***.println(" ");

}

}}

**Output:**



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

**Program:**

public class Quicksort {

String names[]; int length;

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

String stringsList[] = {"hema", "veena", "rani", "gouthami", "honey", "heyaansh", "hello"};

obj.sort(stringsList);

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

}

}

void sort(String array[]) {

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

}

this.names = array; this.length = array.length; quickSort(0, length - 1);

}

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

int j = higherIndex;

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

while (i <= j) {

while (this.names[i].compareToIgnoreCase(pivot) < 0) {

i++;

}

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

}

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

j--;

}

}

if (lowerIndex < j) { quickSort(lowerIndex, j);

}

if (i < higherIndex) { quickSort(i, higherIndex);

}

}

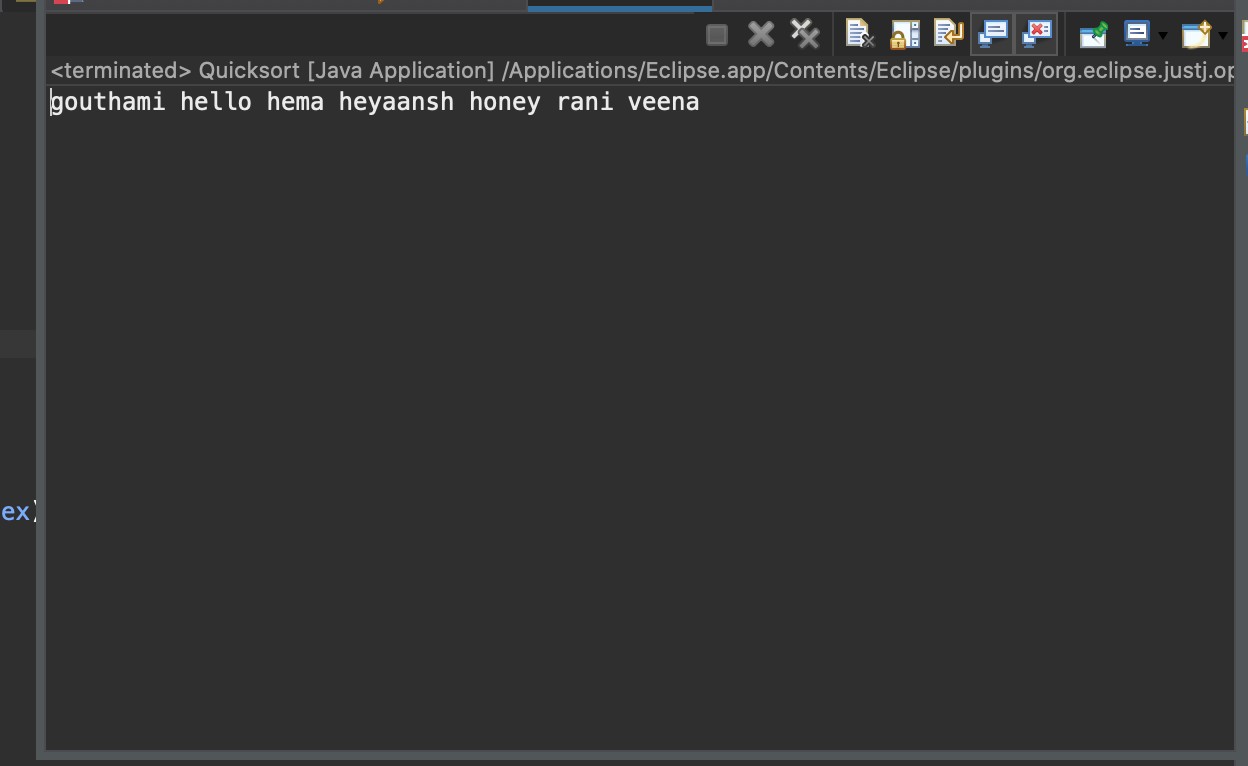
void exchangeNames(int i, int j) {

Sring temp = this.names[i]; this.names[i] = this.names[j]; this.names[j] = temp;

}

}

**Output:**



**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.**

**Program:**

public class BubbleSortExample {

static void bubbleSort(int[] arr) {

int n = arr.length;

int temp = 0;

for(int i=0; i < n; i++){

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

if(arr[j-1] > arr[j]){

//swap elements

temp = arr[j-1];

arr[j-1] = arr[j];

arr[j] = temp;

}

}

}

}

public static void main(String[] args) {

int arr[] ={3,60,35,2,45,320,5};

System.out.println("Array Before Bubble Sort");

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

System.out.print(arr[i] + " "); }

System.out.println();

bubbleSort(arr);//sorting array elements using bubble sort

System.out.println("Array After Bubble Sort");

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

System.out.print(arr[i] + " "); } } }

**Output:**

Array Before Bubble Sort

87 56 4 3 98 0 1

Array After Bubble Sort

98 87 56 4 3 1 0