



JAWAHARLAL COLLEGE OF ENGINEERING AND TECHNOLOGY

JAWHAR GARDENS, LAKKIDI, MAGALAM (PO), PALAKKAD



CSL203 – OBJECT ORIENTED PROGRAMMING LAB (IN JAVA)

LABORATORY MANUAL

B.TECH - CSE

SEMESTER-III

(2019 Regulation-KTU)

Department of Computer Science & Engineering



**JAWAHARLAL COLLEGE OF ENGINEERING &
TECHNOLOGY**

**Approved by AICTE - ISO 9001:2015 Certified, Affiliated to APJ Abdul Kalam
Technological University, Kerala**

**Jawahar Gardens, Lakkidi, Mangalam, Palakkad District, Ottapalam, Kerala –
679301**



JAWAHARLAL COLLEGE OF ENGINEERING AND TECHNOLOGY

JAWHAR GARDENS, LAKKIDI, MAGALAM (PO), PALAKKAD



INSTITUTE VISION

Emerge as a centre of excellence for professional education to produce high quality engineers and entrepreneurs for the development of the region and the Nation.

INSTITUTE MISSION

- To become an ultimate destination for acquiring latest and advanced knowledge in the multidisciplinary domains.
- To provide high quality education in engineering and technology through innovative teaching-learning practices, research and consultancy, embedded with professional ethics.
- To promote intellectual curiosity and thirst for acquiring knowledge through outcome based education.
- To have partnership with industry and reputed institutions to enhance the employability skills of the students and pedagogical pursuits.
- To leverage technologies to solve the real life societal problems through community services.



JAWAHARLAL COLLEGE OF ENGINEERING AND TECHNOLOGY

JAWHAR GARDENS, LAKKIDI, MAGALAM (PO), PALAKKAD



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION

To produce competent professionals with research and innovative skills, by providing them with most conducive environment for quality academic and research oriented undergraduate and postgraduate education along with moral values committed to build a vibrant nation.

MISSION

- Provide a learning environment to develop creativity and problem solving skills in a professional manner.
- Expose to latest technologies and tools used in the field of computer science.
- Provide a platform to explore the industries to understand the work culture and expectation of an organization.
- Enhance Industry Institute Interaction program to develop the entrepreneurship skills.
- Develop research interest among students which will impart a better life for the society and the nation.



JAWAHARLAL COLLEGE OF ENGINEERING AND TECHNOLOGY

JAWHAR GARDENS, LAKKIDI, MAGALAM (PO), PALAKKAD



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

The Graduates in Computer Science and Engineering will be able to:

PEO1: Provide high quality knowledge in computer science and engineering required for a computer professional to identify and solve problems in various application domains.

PEO2: Persist with the ability in innovative ideas in computer support systems and transmit the knowledge and skills for research and advanced learning.

PEO3: Manifest the motivational capabilities and turn on social and economic commitment to community services.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: Use fundamental knowledge of mathematics to solve problems using suitable analysis methods, data structure and algorithms.

PSO2: Interpret the basic concepts and methods of computer systems and technical specifications to provide accurate solutions.

PSO3: Apply theoretical and practical proficiency with a wide area of programming knowledge, design new ideas and innovations towards research.

The following are the Program Outcomes of Engineering Students:

PO1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

CO1	Implement the Object Oriented concepts - constructors, inheritance, method overloading & overriding and polymorphism in Java.
CO2	Implement programs in Java which use datatypes, operators, control statements, built in packages & interfaces, Input / Output streams and Files.
CO3	Implement robust application programs in Java using exception handling.
CO4	Implement application programs in Java using multithreading and database connectivity.
CO5	Implement Graphical User Interface based application programs by utilizing event handling features and Swing in Java.

Lab Manual

PROGRAM 1: WRITE A JAVA PROGRAM THAT CHECKS WHETHER A GIVEN STRING IS A PALINDROME OR NOT

```
import java.util.Scanner;
class Test{
public static void main(String args[]){
Scanner sc = new Scanner(System.in);
System.out.print("Enter the String:");
String str = sc.nextLine();
int flag = 0;
int len = str.length();
for(int i=0;i<len/2;i++){
if(str.charAt(i) != str.charAt(len-i-1)){
flag = 1;
break;
}
}
if(flag == 0){
System.out.println("Palindrome");
}
else{
```



```
System.out.println("Not Palindrome");
}
}
}
```

PROGRAM 2: WRITE A JAVA PROGRAM TO FIND THE FREQUENCY OF A GIVEN CHARACTER IN A STRING

```
import java.util.Scanner;
class Test{
public static void main(String args[]){
Scanner sc = new Scanner(System.in);
System.out.print("Enter the String:");
String str = sc.nextLine();
System.out.print("Enter the character:");
char ch = sc.nextLine().charAt(0);
int count = 0;
for(int i=0;i<str.length();i++){
if(str.charAt(i) == ch){
count++;
}
}
System.out.println("Count of occurrence of "+ ch +"="+count);
}
}
```

PROGRAM 3: WRITE A JAVA PROGRAM TO MULTIPLY TWO GIVEN MATRICES.

```
import java.util.Scanner;
class Test{
public static void main(String args[]){
Scanner sc = new Scanner(System.in);
System.out.print("Enter the order - m1:");
int m1 = sc.nextInt();
System.out.print("Enter the order - n1:");
int n1 = sc.nextInt();
System.out.print("Enter the order - m2:");
int m2 = sc.nextInt();
System.out.print("Enter the order - n2:");
int n2 = sc.nextInt();
if(n1 != m2){
System.out.println("Matrix Multiplication not Possible");
return;
}
```



```
}
int A[][] = new int[m1][n1];
int B[][] = new int[m2][n2];
int C[][] = new int[m1][n2];
System.out.println("Read Matrix A");
for(int i=0;i<m1;i++){
for(int j=0;j<n1;j++){
System.out.print("A["+i+""]["+j+""]=");
A[i][j] = sc.nextInt();
}
}
System.out.println("Read Matrix B");
for(int i=0;i<m2;i++){
for(int j=0;j<n1;j++){
System.out.print("B["+i+""]["+j+""]=");
B[i][j] = sc.nextInt();
}
}
for(int i=0;i<m1;i++){
for(int j=0;j<n2;j++){
C[i][j]=0;
for(int k=0;k<n1;k++){
C[i][j] += A[i][k] * B[k][j];
}
}
}
System.out.println("Matrix A");
for(int i=0;i<m1;i++){
for(int j=0;j<n1;j++){
System.out.print(A[i][j]+"\\t");
}
}
System.out.println();
}
System.out.println("Matrix B");
for(int i=0;i<m2;i++){
for(int j=0;j<n2;j++){
System.out.print(B[i][j]+"\\t");
}
}
System.out.println();
}
System.out.println("Matrix C");
for(int i=0;i<m1;i++){
for(int j=0;j<n2;j++){
System.out.print(C[i][j]+"\\t");
}
}
```




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

PROGRAM 4: WRITE A JAVA PROGRAM WHICH CREATES A CLASS NAMED 'EMPLOYEE' HAVING THE FOLLOWING MEMBERS: NAME, AGE, PHONE NUMBER, ADDRESS, SALARY. IT ALSO HAS A METHOD NAMED 'PRINT- SALARY()' WHICH PRINTS THE SALARY OF THE EMPLOYEE. TWO CLASSES 'OFFICER' AND 'MANAGER' INHERITS THE 'EMPLOYEE' CLASS. THE 'OFFICER' AND 'MANAGER' CLASSES HAVE DATA MEMBERS 'SPECIALIZATION' AND 'DEPARTMENT' RESPECTIVELY. NOW, ASSIGN NAME, AGE, PHONE NUMBER, ADDRESS AND SALARY TO AN OFFICER AND A MANAGER BY MAKING AN OBJECT OF BOTH OF THESE CLASSES AND PRINT THE SAME.

```
import java.util.Scanner;
class Employee{
private String name;
private int age;
private String phone;
private String address;
private double salary;
public void printSalary(){
System.out.println("Salary="+ salary);
}
public Employee(String name,int age,String phone,String address,double salary){
this.name = name;
this.age = age;
this.phone = phone;
this.address = address;
this.salary = salary;
}
public void displayEmployee(){
System.out.println("Name = "+name);
System.out.println("Age = "+age);
System.out.println("Phone Number = "+phone);
System.out.println("Address = "+address);
System.out.println("Salary = "+salary);
}
}
class Manager extends Employee{
private String specialization;
private String department;
public Manager(String name,int age,String phone,String address,double salary,
```



JAWAHARLAL COLLEGE OF ENGINEERING AND TECHNOLOGY

JAWHAR GARDENS, LAKKIDI, MAGALAM (PO), PALAKKAD



```
String specialization,String department){
super(name,age,phone,address,salary);
this.specialization = specialization;
this.department = department;
}
public void displayManager(){
displayEmployee();
System.out.println("Specilization =" +specialization);
System.out.println("Department =" +department);
}
}
class Officer extends Employee{
private String specialization;
private String department;
public Officer(String name,int age,String phone,String address,double salary,
String specialization,String department){
super(name,age,phone,address,salary);
this.specialization = specialization;
this.department = department;
}
public void displayOfficer(){
displayEmployee();
System.out.println("Specilization =" +specialization);
System.out.println("Department =" +department);
}
}
class Test{
public static void main(String args[]){
Scanner sc = new Scanner(System.in);
System.out.println("Enter Manager Details");
System.out.print("Name:");
String name = sc.nextLine();
System.out.print("Age:");
int age = sc.nextInt();sc.nextLine();
System.out.print("Phone Number:");
String phone = sc.nextLine();
System.out.print("Address:");
String addr = sc.nextLine();
System.out.print("Salary:");
double salary = sc.nextDouble();sc.nextLine();
System.out.print("Specialization:");
String spec = sc.nextLine();
System.out.print("Department:");
String dept = sc.nextLine();
Manager m = new Manager(name,age,phone,addr,salary,spec,dept);
```



```
m.displayManager();
System.out.println("Enter Officer Details");
System.out.print("Name:");
String name1 = sc.nextLine();
System.out.print("Age:");
int age1 = sc.nextInt();sc.nextLine();
System.out.print("Phone Number:");
String phone1 = sc.nextLine();
System.out.print("Address:");
String addr1 = sc.nextLine();
System.out.print("Salary:");
double salary1 = sc.nextDouble();sc.nextLine();
System.out.print("Specialization:");
String spec1 = sc.nextLine();
System.out.print("Department:");
String dept1 = sc.nextLine();
Officer o = new Officer(name1,age1,phone1,addr1,salary1,spec1,dept1);
o.displayOfficer();
}
}
```

Output

Enter the officer's Detail

Name:Sangeeth

Address:Trivandrum

Specialization:Computer Science

Department:CSE

Age:32

Number:9633566474

Salary:10000

The officer Detail

Name:Sangeeth

Age:32

Number:9633566474

Address:9633566474

Salary:10000.0

Specialization:Computer Science

Department:CSE

Enter the manager's Detail

Name:Manu

Address:Kochi

Specialization:CSE

Department:Computer Science

Age:30

Number:9895881182

Salary:67000



The manager's Detail

Name:Manu

Age:30

Number:9895881182

Address:9895881182

Salary:67000.0

Specialization:CSE

Department:Computer Science

*/

PROGRAM 5: WRITE A JAVA PROGRAM TO CREATE AN ABSTRACT CLASS NAMED SHAPE THAT CONTAINS AN EMPTY METHOD NAMED NUMBEROFSIDES(). PROVIDE THREE CLASSES NAMED RECTANGLE, TRIANGLE AND HEXAGON SUCH THAT EACH ONE OF THE CLASSES EXTENDS THE CLASS SHAPE. EACH ONE OF THE CLASSES CONTAINS ONLY THE METHOD NUMBEROFSIDES() THAT SHOWS THE NUMBER OF SIDES IN THE GIVEN GEOMETRICAL STRUCTURES. (EXERCISE TO UNDERSTAND POLYMORPHISM).

```
abstract class Shape{
public abstract void numberOfSides();
}
class Rectangle extends Shape{
public void numberOfSides(){
System.out.println("Number of Sides = 4");
}
}
class Triangle extends Shape{
public void numberOfSides(){
System.out.println("Number of Sides = 3");
}
}
class Hexagon extends Shape{
public void numberOfSides(){
System.out.println("Number of Sides = 6");
}
}
class Test{
public static void main(String args[]){
Rectangle r = new Rectangle();
Triangle t = new Triangle();
Hexagon h = new Hexagon();
r.numberOfSides();
t.numberOfSides();
h.numberOfSides();
}
```



```
}  
}
```

PROGRAM 6: WRITE A JAVA PROGRAM TO DEMONSTRATE THE USE OF GARBAGE COLLECTOR

```
class Test{  
public void finalize(){  
System.out.println("Object Memory is released");  
}  
public static void main(String args[]){  
Test t = new Test();  
t = null;  
System.gc();  
}  
}
```

PROGRAM 7: WRITE A FILE HANDLING PROGRAM IN JAVA WITH READER/WRITER.

```
import java.io.*;  
class Test{  
public static void main(String args[]){  
try{  
FileReader fin_1 = new FileReader("file1.txt");  
FileReader fin_2 = new FileReader("file2.txt");  
FileWriter fout = new FileWriter("file3.txt");  
int i;  
while((i=fin_1.read()) != -1){  
fout.write(i);  
}  
while((i=fin_2.read()) != -1){  
fout.write(i);  
}  
fin_1.close();  
fin_2.close();  
fout.close();  
}  
catch(IOException e){  
System.out.println(e.getMessage());  
}  
}  
}
```



PROGRAM 8: WRITE A JAVA PROGRAM THAT READ FROM A FILE AND WRITE TO FILE BY HANDLING ALL FILE RELATED EXCEPTIONS.

```
import java.io.*;
class Test{
public static void main(String args[]){
try{
FileReader fin = new FileReader("test.txt");
FileWriter fout = new FileWriter("copy.txt");
int i;
while((i=fin.read()) != -1){
fout.write(i);
}
fin.close();
fout.close();
}
catch(FileNotFoundException e){
System.out.println(e.getMessage());
}
catch(IOException e){
System.out.println(e.getMessage());
}
}
}
```

PROGRAM 9: WRITE A JAVA PROGRAM THAT READS A LINE OF INTEGERS, AND THEN DISPLAYS EACH INTEGER, AND THE SUM OF ALL THE INTEGERS

```
import java.io.*;
class Test{
public static void main(String args[]){
try{
FileReader fin = new FileReader("test.txt");
BufferedReader br = new BufferedReader(fin);
String inp = br.readLine();
int sum =0;
for(String element: inp.split()){
System.out.println(element);
sum = sum + Integer.parseInt(element);
}
System.out.println("Sum="+sum);
fin.close();
}
}
```



```
catch(IOException e){
System.out.println(e.getMessage());
}
}
}
```

PROGRAM 10: WRITE A JAVA PROGRAM THAT SHOWS THE USAGE OF TRY, CATCH, THROWS AND FINALLY.

```
import java.util.Scanner;
class Test{
public static void divide(int a,int b) throws ArithmeticException{
if(b == 0){
throw new ArithmeticException("Divide by zero is not possible");
}
else{
System.out.println("Result = "+a/b);
}
}
public static void main(String args[]){
int x,y;
try{
Scanner sc = new Scanner(System.in);
System.out.println("Enter the value of x and y");
x = sc.nextInt();
sc.nextLine();
y = sc.nextInt();
sc.nextLine();
divide(x,y);
}
catch(ArithmeticException e){
System.out.println(e.getMessage());
}
finally{
System.out.println("End of Program");
}
}
}
```

Program to perform Division

Enter Number-120

Enter Number-0

/ by zero

End of Operation



PROGRAM 11: WRITE A JAVA PROGRAM THAT IMPLEMENTS A MULTI-THREADED PROGRAM WHICH HAS THREE THREADS. FIRST THREAD GENERATES A RANDOM INTEGER EVERY 1 SECOND. 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 RandonThread extends Thread{
public void run(){
Random r = new Random();
for(int i=0;i<20;i++){
int n = r.nextInt(100); // i will get a value between 0 and 100
if(n % 2 == 0){
new Even(n).start();
}
else{
new Odd(n).start();
}
}
}
}
class Even extends Thread{
private int num;
public Even(int num){
this.num = num;
}
public void run(){
System.out.println("Square of "+num+" =" +num*num);
}
}
class Odd extends Thread{
private int num;
public Odd(int num){
this.num = num;
}
public void run(){
System.out.println("Cube of "+num + " =" + num*num*num);
}
}
}
class Test{
public static void main(String args[]){
RandonThread r = new RandonThread();
r.start();
}
}
```


}

PROGRAM 12: WRITE A JAVA PROGRAM THAT SHOWS THREAD SYNCHRONIZATION.

```
class Display{
public synchronized void print(String msg){
System.out.print("[ "+msg);
try{
Thread.sleep(1000);
}
catch(Exception e){
System.out.println(e.getMessage());
}
System.out.println("]");
}
}
class SyncThread extends Thread{
private Display d;
private String msg;
public SyncThread(Display d,String msg){
this.d=d;
this.msg = msg;
}
public void run(){
d.print(msg);
}
}
class Test{
public static void main(String args[]){
Display d = new Display();
SyncThread t1 = new SyncThread(d,"Hello");
SyncThread t2 = new SyncThread(d,"World");
t1.start();
t2.start();
}
}
```

PROGRAM 13: WRITE A JAVA PROGRAM THAT WORKS AS A SIMPLE CALCULATOR. ARRANGE BUTTONS FOR DIGITS AND THE + - * % OPERATIONS



JAWAHARLAL COLLEGE OF ENGINEERING AND TECHNOLOGY

JAWHAR GARDENS, LAKKIDI, MAGALAM (PO), PALAKKAD



PROPERLY. ADD A TEXT FIELD TO DISPLAY THE RESULT. HANDLE ANY POSSIBLE EXCEPTIONS LIKE DIVIDE BY ZERO. USE JAVA SWING.

```
import javax.swing.*;
import java.awt.event.*;
class Calculator extends JFrame implements ActionListener{
private JTextField t1;
private JButton b1;
private JButton b2;
private JButton b3;
private JButton b4;
private JButton b5;
private JButton b6;
private JButton b7;
private JButton b8;
private JButton b9;
private JButton b10;
private JButton b11;
private JButton b12;
private JButton b13;
private JButton b14;
private JButton b15;
private JButton b16;
private Integer res;
private String operation;
public Calculator(){
setLayout(null);
setSize(640,480);
t1 = new JTextField();
t1.setBounds(100,100,200,30);
b1 = new JButton("1");
b1.setBounds(100,140,50,30);
b2 = new JButton("2");
b2.setBounds(150,140,50,30);
b3 = new JButton("3");
b3.setBounds(200,140,50,30);
b4 = new JButton("+");
b4.setBounds(250,140,50,30);
// Third Row
b5 = new JButton("4");
b5.setBounds(100,170,50,30);
b6 = new JButton("5");
b6.setBounds(150,170,50,30);
b7 = new JButton("6");
b7.setBounds(200,170,50,30);
```



JAWAHARLAL COLLEGE OF ENGINEERING AND TECHNOLOGY

JAWHAR GARDENS, LAKKIDI, MAGALAM (PO), PALAKKAD



```
b8 = new JButton("-");
b8.setBounds(250,170,50,30);
// Fourth Row
b9 = new JButton("7");
b9.setBounds(100,200,50,30);
b10 = new JButton("8");
b10.setBounds(150,200,50,30);
b11 = new JButton("9");
b11.setBounds(200,200,50,30);
b12 = new JButton("*");
b12.setBounds(250,200,50,30);
// Fourth Row
b13 = new JButton("/");
b13.setBounds(100,230,50,30);
b14 = new JButton("%");
b14.setBounds(150,230,50,30);
b15 = new JButton("=");
b15.setBounds(200,230,50,30);
b16 = new JButton("C");
b16.setBounds(250,230,50,30);
add(t1);add(b1);add(b2);
add(b3);add(b4);add(b5);
add(b6);add(b7);add(b8);
add(b9);add(b10);add(b11);
add(b12);add(b13);add(b14);
add(b15);add(b16);
b1.addActionListener(this);b2.addActionListener(this);
b3.addActionListener(this);b4.addActionListener(this);
b5.addActionListener(this);b6.addActionListener(this);
b7.addActionListener(this);b8.addActionListener(this);
b9.addActionListener(this);b10.addActionListener(this);
b11.addActionListener(this);b12.addActionListener(this);
b13.addActionListener(this);b14.addActionListener(this);
b15.addActionListener(this);b16.addActionListener(this);
}
public void doAction(String op){
if(operation == null){
operation = op;
res = Integer.parseInt(t1.getText());
t1.setText("");
}
else{
switch(operation){
case "+": res = res + Integer.parseInt(t1.getText());
break;
```



```
case "-": res = res - Integer.parseInt(t1.getText());
break;
case "/": try{
if(t1.getText().equals("0"){
throw new ArithmeticException("Divide by Zero");
}
res = res / Integer.parseInt(t1.getText());
}
catch(ArithmeticException e){
t1.setText(e.getMessage());
operation = null;
res = 0;
}
break;
case "*": res = res * Integer.parseInt(t1.getText());
break;
case "%": res = res % Integer.parseInt(t1.getText());
break;
}
if(op.equals("=")){
t1.setText(res.toString());
res = 0;
operation = null;
}
else{
operation = op;
t1.setText("");
}
}
}
}
public void actionPerformed(ActionEvent e){
if(e.getSource()== b1)
t1.setText(t1.getText()+"1");
else if(e.getSource()== b2)
t1.setText(t1.getText()+"2");
else if(e.getSource()== b3)
t1.setText(t1.getText()+"3");
else if(e.getSource()== b5)
t1.setText(t1.getText()+"4");
else if(e.getSource()== b6)
t1.setText(t1.getText()+"5");
else if(e.getSource()== b7)
t1.setText(t1.getText()+"6");
else if(e.getSource()== b9)
t1.setText(t1.getText()+"7");
```

```
else if(e.getSource()== b10)
t1.setText(t1.getText()+"8");
else if(e.getSource()== b11)
t1.setText(t1.getText()+"9");
else if(e.getSource()== b16){
t1.setText("");
res =0;
operation = null;
}
else if(e.getSource()== b4){
doAction("+");
}
else if(e.getSource()== b8)
doAction("-");
else if(e.getSource()== b12)
doAction("*");
else if(e.getSource()== b13)
doAction("/");
else if(e.getSource()== b14)
doAction("%");
else if(e.getSource()== b15)
doAction("=");
}
public static void main(String args[]){
new Calculator().setVisible(true);
}
}
```

PROGRAM 14: WRITE A JAVA PROGRAM THAT SIMULATES A TRAFFIC LIGHT. THE PROGRAM LETS THE USER SELECT ONE OF THREE LIGHTS: RED, YELLOW, OR GREEN. WHEN A RADIO BUTTON IS SELECTED, THE LIGHT IS TURNED ON, AND ONLY ONE LIGHT CAN BE ON AT A TIME. NO LIGHT IS ON WHEN THE PROGRAM STARTS.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class TrafficLight extends JPanel implements ActionListener{
private JRadioButton r1;
private JRadioButton r2;
private JRadioButton r3;
private Color red_c;
private Color green_c;
private Color orange_c;
```



```
public TrafficLight(){
setBounds(0,0,600,480);
r1 = new JRadioButton("Red");
r2 = new JRadioButton("Green");
r3 = new JRadioButton("Orange");
ButtonGroup group = new ButtonGroup();
r1.setSelected(true);
group.add(r1);
group.add(r2);
group.add(r3);
add(r1);
add(r2);
add(r3);
red_c = Color.red;
green_c = getBackground ();
orange_c = getBackground();
r1.addActionListener(this);
r2.addActionListener(this);
r3.addActionListener(this);
}
public void actionPerformed(ActionEvent e){
if(r1.isSelected() == true){
red_c = Color.red;
green_c = getBackground ();
orange_c = getBackground();
}
else if(r2.isSelected() == true){
red_c = getBackground ();
green_c = Color.green;
orange_c = getBackground();
}
else if(r3.isSelected() == true){
red_c = getBackground ();
green_c = getBackground();
orange_c = Color.orange;
}
repaint();
}
public void paintComponent(Graphics g){
super.paintComponent(g);
g.drawOval(50,50,50,50);
g.drawOval(50,110,50,50);
g.drawOval(50,170,50,50);
g.setColor(red_c);
g.fillOval(50,50,50,50);
}
```



```
g.setColor(orange_c);
g.fillOval(50,110,50,50);
g.setColor(green_c);
g.fillOval(50,170,50,50);
}
}
class Test{
public static void main(String args[]){
JFrame f1 = new JFrame();
f1.setVisible(true);
f1.setSize(600,480);
f1.setLayout(null);
TrafficLight t = new TrafficLight();
f1.add(t);
}
}
```

PROGRAM 15: WRITE A JAVA PROGRAM TO DISPLAY ALL RECORDS FROM A TABLE USING JAVA DATABASE CONNECTIVITY (JDBC).

```
import java.sql.*;
class Test{
public static void main(String args[]){
try{
//Step 1: Register the driver
Class.forName("com.mysql.cj.jdbc.Driver");
// Step 2: Establish the connection
String url = "jdbc:mysql://localhost:3306/test";
String user = "root";
String pass = "test@123";
Connection con = DriverManager.getConnection(url,user,pass);
// Step 3: Creation of Statement
Statement stmt = con.createStatement();
// Step 4: Execution of Query/Sql
String sql = "select * from person";
ResultSet rs = stmt.executeQuery(sql); // select
while(rs.next()){
System.out.println(rs.getString(1)+"\t"+rs.getInt(2));
}
//Step 5: Closing of Database Connection
con.close();
}
catch(Exception e){
System.out.println(e.getMessage());
}
```



```
}  
}  
}
```

PROGRAM 16: WRITE A JAVA PROGRAM FOR THE FOLLOWING:

1) CREATE A DOUBLY LINKED LIST OF ELEMENTS. 2) DELETE A GIVEN ELEMENT FROM THE ABOVE LIST. 3) DISPLAY THE CONTENTS OF THE LIST AFTER DELETION.

```
import java.util.Scanner;  
class LinkedList{  
private Node head;  
class Node{  
private int data;  
private Node left;  
private Node right;  
public Node(int data){  
this.data = data;  
this.left = null;  
this.right = null;  
}  
}  
public void insert(int data){  
Node temp = new Node(data);  
if(head == null){  
head = temp;  
}  
else{  
Node ptr = head;  
while(ptr.right != null){  
ptr = ptr.right;  
}  
ptr.right = temp;  
temp.left = ptr;  
}  
}  
public void delete(){  
int x = head.data;  
head = head.right;  
head.left = null;  
System.out.println("Element "+x +" got deleted");  
}  
public void display(){  
if(head == null)
```



```
System.out.println("List is Empty");
else{
Node ptr = head;
while(ptr != null){
System.out.print(ptr.data +"\t");
ptr = ptr.right;
}
System.out.println();
}
}
}
class Test{
public static void main(String [] args){
LinkedList list = new LinkedList();
Scanner sc = new Scanner(System.in);
String choice = "";
while(!choice.equals("4")){
System.out.print("1. Insert at End \n2. Delete From Front \n3. Display \n4.Exit\n");
System.out.println("Enter the choice:");
choice = sc.nextLine();
switch(choice){
case "1": System.out.print("Enter the number to insert:");
int data = sc.nextInt();
sc.nextLine();
list.insert(data);
System.out.println("Data inserted Successfully");
break;
case "2": list.delete();
break;
case "3": list.display();
break;
case "4": break;
default: System.out.println("Invalid Choice");
}
} } }
```

PROGRAM 17: WRITE A JAVA PROGRAM THAT IMPLEMENTS QUICK SORT ALGORITHM FOR SORTING A LIST OF NAMES IN ASCENDING ORDER.

```
import java.util.Scanner;
class Test{
public static void quickSort(String A[],int p,int r){
if(p<r){
int q = partition(A,p,r);
```

```
quickSort(A,p,q-1);
quickSort(A,q+1,r);
}
}
public static int partition(String A[],int p,int r){
String x = A[r];
int i = p-1;
for(int j=p;j<=r-1;j++){
if(A[j].compareTo(x) <=0){
i = i + 1;
String temp = A[i];
A[i] = A[j];
A[j] = temp;
}
}
String temp = A[i+1];
A[i+1] = A[r];
A[r] = temp;
return i + 1 ;
}
public static void main(String args[]){
Scanner sc = new Scanner(System.in);
System.out.println("Enter the limit:");
int n = sc.nextInt();
sc.nextLine();
String A[] = new String[n];
System.out.println("Enter the names");
for(int i =0;i<n ;i++){
A[i] = sc.nextLine();
}
quickSort(A,0,n-1);
System.out.println("After Quick Sort");
for(int i =0;i<n;i++)
System.out.println(A[i]);
}
}
```

PROGRAM 18: WRITE A JAVA PROGRAM THAT IMPLEMENTS THE BINARY SEARCH ALGORITHM.

```
class Test{
static int index = -1;
public static int binarySearch(int A[],int lb,int ub,int key){
if(lb <= ub){
```



JAWAHARLAL COLLEGE OF ENGINEERING AND TECHNOLOGY

JAWHAR GARDENS, LAKKIDI, MAGALAM (PO), PALAKKAD



```
int mid = (lb + ub)/2;
if(A[mid] == key)
index = mid;
else if(A[mid] > key)
binarySearch(A,lb,mid-1,key);
else
binarySearch(A,mid+1,ub,key);
}
return index;
}
public static void main(String args[]){
int [] A = {1,7,23,45,56,62,67,98};
int key = 100;
int index = binarySearch(A,0,7,key);
if(index == -1)
System.out.println("Element not found");
else
System.out.println("Element found at index "+index);
}
}
```