CL203

Object Oriented Programming in Java Lab

Lab Manual

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");
		}
	}
}
```

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 occurence of "+ch+"="+count);
	}
}
```

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();
}
```

```
        }
}
```

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,
                                        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
*/
```

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();
	}
}
```

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();
	}
}
```

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());
```

```
		}
	}
}
```

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());
		}
	}
}
```

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());
		}
	}
}
```

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

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();
        }

}
```

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();
        }
}
```

13) Write a Java program that works as a simple calculator. Arrange Buttons for digits and the + - * % operations 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);
		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);
        }

}
```

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);
        }
}
```

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());
                }
        }

}
```

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");
                                }
                        }

                }
        }
```

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]);
        }
```

```
}
```

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){
			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);
	}

}
```