

- 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 occurrence 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 long phoneNumber;
    private String address;
    private double salary;

    public void setName(String name){
        this.name = name;
    }
    public void setAge(int age){
        this.age=age;
    }
    public void setPhoneNumber(long phoneNumber){
        this.phoneNumber = phoneNumber;
    }
    public void setAddress(String address){
        this.address = address;
    }
    public void setSalary(double salary){
        this.salary = salary;
    }
}

```

```

        public double printSalary(){
            return salary;
        }
        public String getName(){
            return name;
        }
        public int getAge(){
            return age;
        }
        public String getAddress(){
            return address;
        }
        public long getPhoneNumber(){
            return phoneNumber;
        }
    }

class Officer extends Employee{
    private String specialization;
    private String department;

    public void setSpecialization(String specialization){
        this.specialization = specialization;
    }
    public void setDepartment(String department){
        this.department = department;
    }

    public String getDepartment(){
        return department;
    }
    public String getSpecialization(){
        return specialization;
    }
}

class Manager extends Employee{
    private String specialization;
    private String department;

    public void setSpecialization(String specialization){
        this.specialization = specialization;
    }
    public void setDepartment(String department){
        this.department = department;
    }

    public String getDepartment(){
        return department;
    }
    public String getSpecialization(){

```

```

        return specialization;
    }

}

class Test{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        Officer o = new Officer();
        System.out.println("Enter the officer's Detail");
        System.out.print("Name:");
        o.setName(sc.nextLine());
        System.out.print("Address:");
        o.setAddress(sc.nextLine());
        System.out.print("Specialization:");
        o.setSpecialization(sc.nextLine());
        System.out.print("Department:");
        o.setDepartment(sc.nextLine());
        System.out.print("Age:");
        o.setAge(sc.nextInt());
        System.out.print("Number:");
        o.setPhoneNumber(sc.nextLong());
        System.out.print("Salary:");
        o.setSalary(sc.nextDouble());
        sc.nextLine(); // to skip new Line
        System.out.println("The officer Detail");
        System.out.println("Name:"+o.getName());
        System.out.println("Age:"+o.getAge());
        System.out.println("Number:"+o.getPhoneNumber());
        System.out.println("Address:"+o.getPhoneNumber());
        System.out.println("Salary:"+o.printSalary());
        System.out.println("Specialization:"+o.getSpecialization());
        System.out.println("Department:"+o.getDepartment());

        Manager m = new Manager();
        System.out.println("Enter the manager's Detail");
        System.out.print("Name:");
        m.setName(sc.nextLine());
        System.out.print("Address:");
        m.setAddress(sc.nextLine());
        System.out.print("Specialization:");
        m.setSpecialization(sc.nextLine());
        System.out.print("Department:");
        m.setDepartment(sc.nextLine());
        System.out.print("Age:");
        m.setAge(sc.nextInt());
        System.out.print("Number:");
        m.setPhoneNumber(sc.nextLong());
        System.out.print("Salary:");
        m.setSalary(sc.nextDouble());
        sc.nextLine(); // to skip new Line
    }
}

```

```

        System.out.println("The manager's Detail");
        System.out.println("Name:"+m.getName());
        System.out.println("Age:"+m.getAge());
        System.out.println("Number:"+m.getPhoneNumber());
        System.out.println("Address:"+m.getPhoneNuber());
        System.out.println("Salary:"+m.printSalary());
        System.out.println("Specialization:"+m.getSpecialization());
        System.out.println("Department:"+m.getDepartment());
    }
}
/*
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();
    }
}

```

Write a Java program that shows the usage of try, catch, throws and finally.

```

import java.util.Scanner;
class Test{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        try{
            System.out.println("Program to perform Division");
            System.out.print("Enter Number-1:");
            int a = sc.nextInt();
            System.out.print("Enter Number-2:");
            int b = sc.nextInt();
            int c = a/b;
            System.out.println("Result="+c);
        }
        catch(ArithmeticException e){
            System.out.println(e.getMessage());
        }
        finally{
            System.out.println("End of Operation");
        }
    }
}

```

```
    }  
}
```

Program to perform Division  
Enter Number-120  
Enter Number-20  
/ by zero  
End of Operation

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 EvenThread extends Thread{  
    private int num;  
    public EvenThread(int num){  
        this.num = num;  
    }  
    public void run(){  
        System.out.println("Square of"+ num+"="+num*num);  
    }  
}  
class OddThread extends Thread{  
    private int num;  
    public OddThread(int num){  
        this.num = num;  
    }  
    public void run(){  
        System.out.println("Cube of "+num+"="+ num*num*num);  
    }  
}  
class RandomThread extends Thread{  
    public void run(){  
        Random r = new Random();  
        for(int i =0;i<10;i++){  
            int num = r.nextInt(100);  
            if(num % 2 == 0){  
                new EvenThread(num).start();  
            }  
            else{  
                new OddThread(num).start();  
            }  
        }  
    }  
}  
  
class Test{
```



```

        public static void main(String args[]){
            RandomThread r = new RandomThread();
            r.start();
        }
    }

    /*
    class EvenThread extends Thread{
        private int num;
        public EvenThread(int num){
            this.num = num;
        }
        public void run(){
            System.out.println(num*num);
        }
    }
    */

```

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();
    }
}

```

## Calculator

```

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);
}

}

```

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());
        }
    }
}

```

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

Write a Java program that reads a line of integers, and then displays each integer, and the sum of all the integers

```
import java.util.Scanner;
import java.util.StringTokenizer;
class Test{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the limit");
        int n = sc.nextInt();
        sc.nextLine();

        System.out.println("Enter the "+ n + " number (space seperated)");
        String numList = sc.nextLine();

        StringTokenizer st = new StringTokenizer(numList," ");
        // Tokenization is process of spliting the given sentence.
        int sum = 0;
        while(st.hasMoreTokens()){ //[10,20,30,40,50]
            sum = sum + Integer.parseInt(st.nextToken());
        }
        System.out.println("Sum = "+sum);
    }
}
```

QuickSort

```
class QuickSort{

    public static int partition(String[] A,int p,int r){
```

```

        String temp;
        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;
                temp = A[i];
                A[i] = A[j];
                A[j] = temp;
            }
        }
        temp = A[i+1];
        A[i+1] = A[r];
        A[r] = temp;
        return i+1;
    }

    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 void main(String [] args){
        String names[] ={"Sangeeth","Poornima",
        "Pratheesh","Vikas","Karthi","Abin"};
        quickSort(names,0,5);
        System.out.println("After QuickSort");
        for(int i=0;i<names.length;i++){
            System.out.println(names[i]);
        }
    }
}

```

#### Doubly LinkedList

```

import java.util.Scanner;
class DoublyLinkedList{
    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 insertAtEnd(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;
    }
    System.out.println(data+ " is inserted Successfully");
}

public void deleteAtFront(){

    if(head == null){
        System.out.println("List is Empty");
    }
    else{
        int data = head.data;
        head = head.right;
        head.left = null;
        System.out.println(data+" is deleted from the list");
    }

}

public void display(){
    Node temp = head;
    if(head == null){
        System.out.println("List Empty");
    }
    else{
        while(temp != null){
            System.out.print(temp.data+"\t");
            temp = temp.right;
        }
        System.out.println();
    }
}

}

class Test{
    public static void main(String args[]){
        DoublyLinkedList dl = new DoublyLinkedList();
        int ch = 0;
    }
}

```



```

while(ch != 4){
    System.out.println("1.Insert\n2.Delete\n3.Display\n4.Exit");
    Scanner sc = new Scanner(System.in);
    ch = sc.nextInt();
    sc.nextLine();

    switch(ch){
        case 1: System.out.println("Enter the element");
                int elt = sc.nextInt();
                sc.nextLine();
                dl.insertAtEnd(elt);
                break;
        case 2: dl.deleteAtFront();
                break;
        case 3: dl.display();
                break;
        case 4: break;
        default: System.out.println("Invalid Choice");
    }
}
}
}

```

## Binary Search

```

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);
    }
}

```

## Traffic Light

```
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 yellow_c;

    public TrafficLight(){
        setBounds(0,0,640,480);
        r1 = new JRadioButton("Red");
        r2 = new JRadioButton("Yellow");
        r3 = new JRadioButton("Green");

        r1.setSelected(true);
        red_c = Color.red;
        yellow_c = getBackground();
        green_c = getBackground();

        ButtonGroup gp = new ButtonGroup();
        gp.add(r1);
        gp.add(r2);
        gp.add(r3);

        add(r1);
        add(r2);
        add(r3);

        r1.addActionListener(this);
        r2.addActionListener(this);
        r3.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e){
        if(r1.isSelected() == true){
            red_c = Color.red;
            yellow_c = getBackground();
            green_c = getBackground();
        }
        else if(r2.isSelected() == true){
            red_c = getBackground();
            yellow_c = Color.yellow;
            green_c = getBackground();
        }
    }
}
```

```

        else if(r3.isSelected() == true){
            red_c = getBackground();
            yellow_c = getBackground();
            green_c = Color.green;
        }
        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(yellow_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 f = new JFrame();
        f.setVisible(true);
        f.setSize(640,480);
        f.setLayout(null);
        TrafficLight t = new TrafficLight();
        f.add(t);
    }
}

```