

Java Programs: -

1. A shop will give discount of 10% if the cost of purchased quantity is more than 1000.

Ask user for quantity

Suppose, one unit will cost 100.

Judge and print total cost for u

```
import java.util.*;
class Q1{
    public static void main(String args[]){
        int units;
        double ppu ,total_cost;
        Scanner s = new Scanner(System.in);
        System.out.println("Enter number of units: ");
        units = s.nextInt();
        System.out.println("Enter price of each unit: ");
        ppu = s.nextDouble();
        total_cost= units*ppu;
        if(total_cost>1000){
            total_cost = total_cost*0.9;
        }
        System.out.println("Total final cost: "+total_cost);
    }
}
```

2. Given five positive integers, find the minimum and maximum values that can be calculated by summing exactly four of the five integers. Then print the respective minimum and maximum values as a single line of two space-separated long integers.

Example 1 2 3 4 5

The minimum sum is 1+2+3+4=10 and the maximum sum is 2+3+4+5=14. The function prints

```
import java.util.*;
class Q2{
    public static void main(String args[]){
        int i, j, min=0, max=0, n, temp;
        int a[] = new int[50];
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of elements: ");
        n = sc.nextInt();
        for(i=0;i<n;i++){
            System.out.print("Enter element "+(i+1)+" : ");
            a[i] = sc.nextInt();
        }
        for(i=0;i<n-1;i++){
            for(j=0;j<n-1;j++){
                if(a[j]>a[j+1]){
                    temp = a[j];
                    a[j] = a[j+1];
                    a[j+1] = temp;
                }
            }
        }
    }
}
```

```

    }
}
for(i=0;i<n-1;i++){
    min= min+ a[i];
}
for(i=1;i<n;i++){
    max= max+ a[i];
}
System.out.println("Minimum sum: "+min);
System.out.println("Maximum sum: "+max);
sc.close();
}
}

```

3. Implement a java program to calculate gross salary and net salary taking the following data.

Input: empno, empname, basic

Process

DA=70% of basic

HRA=30% of basic

CCA= Rs. 240/-

PF=10% of basic

PT=Rs.100/-

```

import java.util.*;
class Q3{
    public static void main(String[] args) {
        double da,hra,cca,pt,pf,gross,net,basic;
        Scanner sc = new Scanner(System.in);
        String empname = new String();
        String empno = new String();
        System.out.print("Enter Employee Name: ");
        empname = sc.nextLine();
        System.out.print("Enter Employee Number: ");
        empno = sc.next();
        System.out.print("Enter Basic: ");
        basic = sc.nextDouble();
        da = 0.7*basic;
        hra = 0.3*basic;
        cca=240;
        pf=0.1*basic;
        pt=100;
        gross = basic+da+hra;
        net = gross-pt-cca-pf;
        System.out.println("Gross salary = "+gross+"\nNet salary=
"+net);
    }
}

```

4. Write a menu driven program to perform the following operations on one

- dimensional array. (Accept data from user)
- a) to search a given element in an array
 - b) to sort elements in descending order
 - c) to delete the duplicate elements from an array

```
import java.util.*;
class Q4 {
    public static void main(String[] args) {
        int i,j,n,temp,choice;
        int a[]= new int[20];
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of elements: ");
        n = sc.nextInt();
        for(i=0;i<n;i++){
            a[i]= sc.nextInt();
        }
        System.out.print("MENU\n1. Search element\n2. Sort
descending\n3. Delete Duplicate\nEnter your choice: ");
        choice = sc.nextInt();
        switch(choice){
            case 1:
                int flag=0,ele;
                System.out.println("Enter element to be searched: ");
                ele = sc.nextInt();
                for(i=0;i<n;i++){
                    if(a[i]==ele){
                        flag=1;
                        break;
                    }
                }
                if(flag==1){
                    System.out.println(ele+" has been found!");
                }
                else{
                    System.out.println(ele+" not found!");
                }
                break;
            case 2:
                for(i=0;i<n-1;i++){
                    for(j=0;j<n-1;j++){
                        if(a[j]<a[j+1]){
                            temp = a[j];
                            a[j] = a[j+1];
                            a[j+1] = temp;
                        }
                    }
                }
                for(i=0;i<n;i++){
                    System.out.print(a[i]+" ");
                }
            }
    }
}
```

```

        break;
    case 3:
        int tem[] = new int[n];
        int count=0;
        for(i=0;i<n-1;i++){
            for(j=0;j<n-1;j++){
                if(a[j]>a[j+1]){
                    temp = a[j];
                    a[j] = a[j+1];
                    a[j+1] = temp;
                }
            }
        }
        for(i=0;i<n-1;i++){
            if(a[i]!=a[i+1]){
                tem[count++]=a[i];
            }
        }
        tem[count++] = a[n - 1];

        for(i=0;i<count;i++){
            System.out.print(tem[i]+" ");
        }
        break;
    default:
        System.out.println("Enter a number between 1-3.");
    }
    sc.close();
}
}

```

5. An Election is contested by 5 candidates. The candidate numbers are 1 to 5 and the voting is done by making the candidate number on the ballot paper. Write a program to read the ballots and count the vote for each candidate using an array variable count. In case, a number read is outside the range 1 to 5, the ballot should be considered as a 'spoilt ballot' and the program should also count the number of spoiled ballots.

```

import java.util.*;
class Q5 {
    public static void main(String[] args) {
        int i,n,j;
        int a[] = new int[50];
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        for(i=0;i<n;i++){
            System.out.print((i+1)+" Vote: ");

```

```

        a[i]=sc.nextInt();
    }
    int c1=0,c2=0,c3=0,c4=0,c5=0,sv=0;
    for(i=0;i<n;i++){
        if(a[i]==1){
            c1+=1;
        }
        else if(a[i]==2){
            c2+=1;
        }
        else if(a[i]==3){
            c3+=1;
        }
        else if(a[i]==4){
            c4+=1;
        }
        else if(a[i]==5){
            c5+=1;
        }
        else{
            sv+=1;
        }
    }
    System.out.println(c1+" "+c2+" "+c3+" "+c4+" "+c5+" "+sv);
    sc.close();
}
}

```

6. Given an unsorted integer array, find the first missing positive integer. Example,
 If A = [-1, 4, 2, 3, 5], missing integer = 1.
 If A = [1, 5, 2, 3, 4, 7], missing integer = 6.
 If A = [-1, -2, -3, -4], missing integer = 1.

```

import java.util.*;
class Q6 {
    public static void main(String[] args) {
        int i,n,j,temp;
        int a[] = new int[50];
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        for(i=0;i<n;i++){
            System.out.print((i+1)+" Value: ");
            a[i]=sc.nextInt();
        }
        for(i=0;i<n-1;i++){
            for(j=0;j<n-1;j++){
                if(a[j]>a[j+1]){
                    temp = a[j];
                    a[j] = a[j+1];

```

```

        a[j+1] = temp;
    }
}
}
int minm= a[0];
for(i=0;i<n;i++){
    if (a[i]!=minm){
        System.out.println("Missing value is: "+minm);
        break;
    }
    minm++;
}
sc.close();
}
}

```

7. Caesar cipher is one of the simplest encryption techniques. By using this cipher technique, we can replace each letter in the plaintext with different one a fixed number of places up or down the alphabet.
For example:

With right shift of 3:

plaintext: ABCDEFGHIJKLMNOPQRSTUVWXYZ

ciphertext: DEFGHIJKLMNOPQRSTUVWXYZABC

```

class Q7 {
    public static void main(String[] args) {
        int i,j,asc;
        char temp;
        String s = new String("WXYZ");
        String c = new String("");
        for(i=0;i<s.length();i++){
            temp=s.charAt(i);
            asc = (int)(temp);
            asc = 65 + (asc-65+3)%26;
            // System.out.print(asc);

            temp = (char)(asc);
            c+=temp;
        }
        System.out.println(c);
    }
}

```

8. Write a constructor in the Car class given below that initializes the brand class field with the string "Ford." Call the getBrand () method in the main method of the Sample class and store the value of the brand in a variable, and print the value.

```

class Car {
    String brand = new String();
    Car(){
        this("Ford");
    }
    Car(String br){

        this.brand =br;
    }
    void getbrand(){
        System.out.println("The car brand is "+this.brand);
    }
}
class Q8{
    public static void main(String[] args) {
        Car c = new Car();
        c.getbrand();
        Car d = new Car("BMW");
        d.getbrand();
    }
}

```

9. Write a java program to create a user-defined package letmecalculate having class calculator and functions addition, subtraction, multiplication, division. Import this package in another program to use the class calculator.

letmecalculate/calculator.java

```

package letmecalculate;
public class calculator{
    public double add(double x,double y){
        return x+y;
    }
    public double sub(double x,double y){
        return x-y;
    }
    public double mul(double x,double y){
        return x*y;
    }
    public double div(double x,double y){
        return x/y;
    }
}

```

q9.java

```

import letmecalculate.calculator;
class Q9 {
    public static void main(String[] args) {
        double a=9.2,b=8.8;
    }
}

```

```

        calculator c = new calculator();
        System.out.println(c.add(a,b));
        System.out.println(c.sub(a,b));
        System.out.println(c.mul(a,b));
        System.out.println(c.div(a,b));
    }
}

```

10. WAP to find whether the entered 4-digit number is vampire or not. Combination of digits from this number forms 2-digit number. When they are multiplied by each other we get the original number. (1260=21*60, 1395=15*93, 1530=30*51)

```

import java.util.*;
class Q10{
    public static void main(String[] args) {
        int n,i=0,j,temp,k,l;
        int a[] = new int[4];
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter 4 dig no: ");
        n = sc.nextInt();
        int og=n;
        while(n>0){
            temp = n%10;
            a[i++]=temp;
            n=n/10;
        }
        int num1,num2;
        for(i=0;i<4;i++){
            for(j=0;j<4;j++){
                if(i!=j){
                    num1=10*a[i]+a[j];
                    for(k=0;k<4;k++){
                        if(k!=i && k!=j){
                            for(l=0;l<4;l++){
                                if(l!=i && l!=j && l!=k){
                                    num2 = 10*a[k]+a[l];
                                    if(num1*num2==og){
                                        System.out.println(og+" is a
Vampire number with solution "+num1+" , "+num2);
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
        sc.close();
    }
}

```



```
}
}
```

11. Write a java program to count number of alphabets, digits, special symbols, blank spaces, and words from the given sentence. Also count number of vowels and consonants

```
class Q11 {
    public static void main(String[] args) {
        String s = new String("Dev scored 10.00 in semester 4!");
        int i, al=0, no=0, sc=0, sp=0, vow=0, con=0;
        char c;
        for(i=0; i<s.length(); i++){
            s = s.toLowerCase();
            c = s.charAt(i);
            if(c>='0' && c<='9'){
                no+=1;
            }
            else if(c>='a' && c<='z'){
                al+=1;
                if(c=='a' || c=='e' || c=='i' || c=='o' || c=='u'){
                    vow+=1;
                }
                else{
                    con+=1;
                }
            }
            else if(c==' '){
                sp+=1;
            }
            else{
                sc+=1;
            }
        }
        System.out.println("Alphabet "+al+"\nvowels "+vow+"\nconso
"+con+" \ndigits "+no+"\nspace "+ sp+ " \nwords "+(sp+1)+"\nschar
"+sc);
    }
}
```

12. WAP that accepts a shopping list of items and performs the following operations: Add an item at a specified location, delete an item in the list, and print the contents of the vector

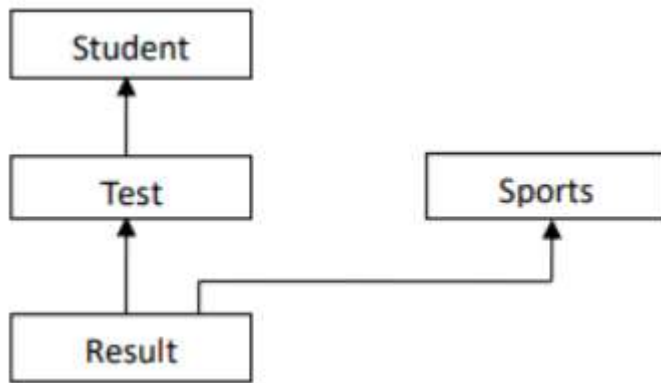
```
import java.util.*;
class Q12 {
    public static void main(String[] args) {
        Vector sl = new Vector(3, 2);
        Scanner sc = new Scanner(System.in);
```

```

        String ele = new String();
        while(true){
            System.out.print("\nMENU\n1.Add element to pos\n2.Delete an
item\n3.Print list\n4.Exit\nEnter your choice:");
            int ch = sc.nextInt();
            switch(ch){
                case 1:
                    System.out.print("\nEnter position: ");
                    int pos=sc.nextInt();
                    System.out.print("\nEnter element: ");
                    ele=sc.next();
                    if(pos<=sl.size()){
                        sl.insertElementAt(ele, pos-1);
                    }
                    else{
                        sl.addElement(ele);
                    }
                    break;
                case 2:
                    System.out.print("\nEnter element: ");
                    ele=sc.next();
                    sl.removeElement(ele);
                    break;
                case 3:
                    System.out.println("\nCurrent list: ");
                    System.out.println(sl);
                    break;
                case 4:
                    System.exit(0);
            }
        }
        // sc.close();
    }
}

```

13. WAP to implement three classes namely Student, Test and Result. Student class has member as rollno, Test class has members as sem1_marks and sem2_marks and Result class has member as total. Create an interface named sports that has a member score (). Derive Test class from Student and Result class has multiple inheritances from Test and Sports. Total is formula based on sem1_marks, sem2_mark and score.



```

class Q13 {
    public static void main(String[] args) {
        Result r = new Result(16, 0, 95, 90);
        r.score();
    }
}

class Student{
    int rollno;
}

class Test extends Student{
    int sem1_marks, sem2_marks;
}

interface Sports{
    void score();
}

class Result extends Test implements Sports{
    int total,sports_score;
    Result(int rn, int sc, int s1, int s2){
        this.rollno = rn;
        this.sports_score = sc;
        this.sem1_marks=s1;
        this.sem2_marks=s2;
        this.total = sc+s1+s2;
    }
    public void score(){
        System.out.println("Roll No: "+rollno+"\nSem1 Marks:
"+sem1_marks+"\nSem2 Marks: "+sem2_marks+"\nSports Score:
"+sports_score);
        System.out.println("Total score: "+total);
    }
}

```

14. Create the Account class Account.java and write a main method in a different class to briefly experiment with some instances of the Account class. Using the Account class

as a base class, write two derived classes called SavingsAccount and CurrentAccount. A SavingsAccount object, in addition to the attributes of an Account object, should have an interest variable and a method which adds interest to the account. A CurrentAccount object, in addition to the attributes of an Account object, should have an overdraft limit variable. Ensure that you have overridden methods of the Account class as necessary in both derived classes. Now create a Bank class, an object of which contains an array of Account objects. Accounts in the array could be instances of the Account class, the SavingsAccount class, or the CurrentAccount class. Create some test accounts (some of each type). Write an update method in the bank class. It iterates through each account, updating it in the following ways: Savings accounts get interest added (via the method you already wrote); CurrentAccounts get a letter sent if they are in overdraft. The Bank class requires methods for opening and closing accounts, and for paying a dividend into each account. Hints: Note that the balance of an account may only be modified through the deposit (double) and withdraw (double) methods.

The Account class should not need to be modified at all. Be sure to test what you have done after each step.

```
class Q14 {
    public static void main(String[] args) {
        CurrentAccount acc1 = new CurrentAccount("A", "x", 5000, 2000);
        SavingsAccount acc2 = new SavingsAccount("B", "y", 8000);
        acc1.withdraw();
        acc1.getbalance();
        acc2.addint();
        acc2.getbalance();
    }
}

class Account{
    String name, accno;
    double balance;
    Account(String n, String acc, double bal){
        this.name = n;
        this.accno = acc;
        this.balance=bal;
    }
    void getbalance(){
        System.out.println(balance);
    }
}

class SavingsAccount extends Account{
    double interest=0.065;
    SavingsAccount(String n, String acc, double bal){
        super(n, acc, bal);
    }
    void addint(){
        balance=balance+balance*interest;
    }
}
```

```

    }
}

class CurrentAccount extends Account{
    int odlimit=6000,wd;
    CurrentAccount(String n, String acc, double bal,int w){
        super(n,acc,bal);
        this.wd=w;
    }
    void withdraw(){
        if(wd>odlimit || wd>balance){
            System.out.println("Aukaat ke bahar hai");
        }
        else{
            balance=balance-wd;
        }
    }
}
}

```

15. Write a Java Program to calculate the Result. Result should consist of name, seatno, date, center number and marks of semester three exam. Create a User Defined Exception class MarksOutOfBoundsException, If Entered marks of any subject is greater than 100 or less than 0, and then program should create a user defined Exception of type MarksOutOfBoundsException and must have a provision to handle it

```

class Q15 {
    public static void main(String[] args) {
        // int mrk[] = new int[]{94,67,-3,109,5};
        Result r = new Result("Dev Patel", 16, "12/12/2022", 4, 109 ,
5);
        r.giveResult();
    }
}

class MarksOutOfBoundsException extends Exception{
    public MarksOutOfBoundsException(String s){
        System.out.println(s);
    }
}

class Result{

    int seatno, cno,n,marks;
    String date = new String();
    String name = new String();

    Result(String nm, int s, String d, int center, int m, int no){
        this.name= nm;
        this.date=d;
    }
}

```

```

        this.seatno=s;
        this.cno=center;
        this.marks=m;
        this.n=no;
    }
    void giveResult(){
        System.out.println("Name: "+name);
        System.out.println("Date: "+date);
        System.out.println("Seat No: "+seatno);
        System.out.println("Center No: "+cno);
        try{
            if(marks>100 || marks<0){
                throw new MarksOutOfBoundsException("Exception: Marks
out of range");
            }
            else{System.out.println("Marks for subject: "+marks);}
        }
        catch(MarksOutOfBoundsException e){
            System.out.print(e);
        }
    }
}

```

16. Write a Java Program to input the data through command Line and Find out total valid and in-valid integers. (Hint: use exception handling)

```

import java.util.Scanner;
class Q16 {
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string of characters: ");
        String x = sc.nextLine();
        int validCount = 0;
        int invalidCount = 0;
        for (int i = 0; i < x.length(); i++) {
            try{
                int num = Integer.parseInt(
Character.toString(x.charAt(i)));
                validCount+=1;
            }catch(NumberFormatException e){
                invalidCount+=1;
            }
        }
        System.out.println("valid integers : " + validCount);
        System.out.println("Invalid Integers : " + invalidCount);
        sc.close();
    }
}

```

```
}
```

17. Write java program to implement the concept of Thread Synchronization (Ticket Booking Example)

```
class Q17 {
    public static void main(String[] args) {
        Booking d = new Booking(1);
        Thread t1 = new Thread(d);
        Thread t2 = new Thread(d);
        t1.setName("Dev");
        t2.setName("Alistair");
        t2.start();
        t1.start();
    }
}

class Booking extends Thread{
    int vacant=1, reqd;
    Booking(int r){
        this.reqd = r;
    }
    public synchronized void run(){
        if(reqd<=vacant){
            System.out.println(reqd+ " ticket(s) booked for "
            +Thread.currentThread().getName());
            try{
                Thread.sleep(100);
            }
            catch(Exception e){
            }
            vacant = vacant-reqd;
        }
        else{
            System.out.println("No tickets left. Sorry, "
            +Thread.currentThread().getName());
        }
    }
}
```

Scala Problem Statements

18. Write a program to print given no in words using pattern matching and while loop .eg
123 output one two three.

```
object Main{
    def main(args: Array[String]):Unit={
        var n = scala.io.StdIn.readInt();
        val og =n;
```

```

var s:String=""
while(n>0){
    var temp = n%10;
    n= n/10;
    temp match{
        case 1=> s="One "+s
        case 2=> s="Two "+s
        case 3=> s="Three "+s
        case 4=> s="Four "+s
        case 5=> s="Five "+s
        case 6=> s="Six "+s
        case 7=> s="Seven "+s
        case 8=> s="Eight "+s
        case 9=> s="Nine "+s
        case 0=> s="Zero "+s
        case _=> s="Non-digit "+s
    }
}
print(s)
}

```

19. Write a program to find whether the no is prime or not using do while loop.

```

object q19{
    def main(args: Array[String]):Unit={
        val n = scala.io.StdIn.readInt();
        var i=2;
        var flag=0;
        do{
            if(n%i==0){
                flag=1;
            }
        }while(i<=n);
    }
}

```

20. Write a program in Scala to demonstrate string interpolation.

```

object Main{
    def main(args: Array[String]):Unit={
        var name: String = "Dev Patel";
        var sap:String="60009200016"
        print(s"My name is $name and my SAP ID is $sap")
    }
}

```

21. Create a class employee with data member empid, empname, designation and salary. Write a methods get_employee()-to take user input, show_grade –to display grade of the employee based on salary.

Show employee () to display employee details.

4. Salary Range	5. Grade
6. <10000	7. D
8. 10000-24999	9. C
10. 25000-49999	11. B
12. >50000	13. A

```
object Main{
  def main(args: Array[String]):Unit={
    var e = new Employee()
    e.getEmployee();
    e.showGrade();
    e.showEmployee();
  }
}

class Employee{
  var empid:Int=0;
  var empname:String=null;
  var designation:String=null;
  var salary:Int=0;
  var grade:Char='-';
  def getEmployee(){
    print("Enter Emp ID: ");
    this.empid = scala.io.StdIn.readInt();
    print("Enter Emp Name: ");
    this.empname = scala.io.StdIn.readLine();
    print("Enter Designation: ");
    this.designation= scala.io.StdIn.readLine();
    print("Enter Salary: ");
    this.salary = scala.io.StdIn.readInt();
  }
  def showGrade(){
    if(this.salary<10000){
      this.grade='D';
      println("Employee Grade: "+this.grade);
    }
    else if(this.salary>=10000 && this.salary<25000){
      this.grade='C';
      println("Employee Grade: "+this.grade);
    }
    else if(this.salary>=25000 && this.salary<50000){
      this.grade='B';
      println("Employee Grade: "+this.grade);
    }
  }
}
```

```

    else{
        this.grade='A';
        println("Employee Grade: "+this.grade);
    }

}

def showEmployee(){
    print(this.empid
,this.empname,this.designation,this.salary,this.grade)
}
}

```

22. Five Bikers Compete in a race such that they Drive at Constant speed which may or may not be same as the other. To qualify the race, the speed of as racer must be more than the average speed of all 5 racers. Write Scala program to take as an input the speed of all racer and print back the speed of qualifying racer.

```

object Main{
    def main(args: Array[String]):Unit={
        var r1:Int=0;
        var r2:Int=0;
        var r3:Int=0;
        var r4:Int=0;
        var r5:Int=0;
        print("Enter speed of racer 1: ");
        r1=scala.io.StdIn.readInt();
        print("Enter speed of racer 2: ");
        r2=scala.io.StdIn.readInt();
        print("Enter speed of racer 3: ");
        r3=scala.io.StdIn.readInt();
        print("Enter speed of racer 4: ");
        r4=scala.io.StdIn.readInt();
        print("Enter speed of racer 5: ");
        r5=scala.io.StdIn.readInt();
        var avg:Double = (r1+r2+r3+r4+r5)/5;
        print(avg)
        if(r1>avg){
            print("Racer 1 qualifies with speed "+r1)
        }
        if(r2>avg){
            print("Racer 2 qualifies with speed "+r2)
        }
        if(r3>avg){
            print("Racer 3 qualifies with speed "+r3)
        }
        if(r4>avg){
            print("Racer 4 qualifies with speed "+r4)
        }
        if(r5>avg){
            print("Racer 5 qualifies with speed "+r5)
        }
    }
}

```

```

    }
}
}

```

23. Create a class named 'Member' having the following members:

Data members

- 1 - Name
- 2 - Age
- 3 - Phone number
- 4 - Address
- 5 - Salary

It also has a method named 'printSalary' which prints the salary of the members. Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same.

```

object Main{
  def main(args: Array[String]){
    var e = new Employee();
    e.getMember();
    e.printSalary();
    e.showMember();
    var m = new Manager();
    m.getMember();
    m.printSalary();
    m.showMember();
  }
}

class Member{
  var name:String = null;
  var age:Int=0;
  var phoneno:String=null;
  var address:String=null;
  var salary:Int=0;
  def getMember():Unit={
    print("Name: ")
    this.name = scala.io.StdIn.readLine();
    print("Age: ")
    this.age = scala.io.StdIn.readInt();
    print("Phone no: ")
    this.phoneno = scala.io.StdIn.readLine();
    print("Address: ")
    this.address = scala.io.StdIn.readLine();
    print("Salary: ")
    this.salary = scala.io.StdIn.readInt();
  }
  def showMember(){

```

```

        println("Name: "+this.name)
        println("Age: "+this.age)
        println("Phone no: "+this.phoneno)
        println("Address: "+this.address)
        println("Salary: "+this.salary)
    }
    def printSalary():Unit={
        println(this.salary)
    }
}

class Employee extends Member{
    var specialization:String=null;
    override def getMember(){
        super.getMember
        print("Specialization: ")
        this.specialization = scala.io.StdIn.readLine();
    }
    override def showMember(){
        super.showMember
        print("Specialization: "+this.specialization)
    }
}

class Manager extends Member{
    var department:String=null;
    override def getMember(){
        super.getMember
        print("Department: ")
        this.department = scala.io.StdIn.readLine();
    }
    override def showMember(){
        super.showMember
        print("Department: "+this.department)
    }
}

```

24. Design a class hierarchy rooted in the class Employee that includes subclasses for HourlyEmployee and SalaryEmployee. The attributes shared in common by these classes include the name, and job title of the employee, plus the accessor and mutator methods needed by those attributes. The salaried employees need an attribute for weekly salary, and the corresponding methods for accessing and changing this variable. The hourly employees should have a pay rate and an 'hoursworked' variable. There should be an abstract method called calculateWeeklyPay (), defined abstractly in the superclass and implemented in the subclasses. The salaried worker's pay is just the weekly salary. Pay for an hourly employee is simply hours worked times pay rate.

```

object Main{
    def main(args:Array[String]):Unit={
        var h = new HourlyEmployee();
    }
}

```

```

        h.accessor();
        h.calculateWeeklyPay();
        h.mutator();
        h.calculateWeeklyPay();

        var w = new WeeklyEmployee();
        w.accessor();
        w.calculateWeeklyPay();
        w.mutator();
        w.calculateWeeklyPay();
    }
}

abstract class Employee{
    var name:String= null;
    var jobtitle:String = null;
    def accessor();
    def mutator();
    def calculateWeeklyPay();
}

class HourlyEmployee extends Employee{
    var payrate:Int=0;
    var hoursworked:Int=0;
    def accessor(){
        println("Enter payrate: ")
        payrate = scala.io.StdIn.readInt();
        println("Enter Hours worked this week: ")
        hoursworked = scala.io.StdIn.readInt();
    }
    def mutator(){
        println("Enter changed payrate: ")
        payrate = scala.io.StdIn.readInt();
        println("Enter the changed hours worked this week: ")
        hoursworked = scala.io.StdIn.readInt();
    }
    def calculateWeeklyPay(){
        println("Weekly pay is: "+(this.payrate*this.hoursworked));
    }
}

class WeeklyEmployee extends Employee{
    var weekllysalary:Int=0;
    def accessor(){
        println("Enter Weekly Salary: ")
        weekllysalary = scala.io.StdIn.readInt();
    }
    def mutator(){

```

```

println("Enter changed Weekly Salary: ")
weekllysalary = scala.io.StdIn.readInt();
}
def calculateWeeklyPay(){
println("Weekly pay is: "+this.weekllysalary);
}
}

```

25. Write a recursive function to get the nth Fibonacci number. The first two Fibonacci numbers are 0 and 1. The nth number is always the sum of the previous two—the sequence begins 0, 1, 1, 2, 3, 5. `def fib (n: Int): Int`

```

object Main{
def main(args: Array[String]){
println("Enter number: ")
var n:Int=scala.io.StdIn.readInt();
var t1:Int=0;
var t2:Int=1;
var fib:Int=fibonacci(t1,t2,n-2);
println(fib)
}
def fibonacci(t1:Int,t2:Int,n:Int):Int={
if(n==0){
return t2
}
fibonacci(t2,t1+t2,n-1);
}
}

```

26. Write a function to find the values of following series: -Value= $a+a^2/2! +a^3/3! +a^4/4! + \dots + a^n/n!$ (Use passing function as parameter to another function)

```

object Main{
def main(args:Array[String]):Unit={
// var a:Int=scala.io.StdIn.readInt();
// var n:Int=scala.io.StdIn.readInt();
func(2,3,func2);
}
def func(a:Int,n:Int,f:(Int,Int)=>Unit):Unit={
f(a,n);
}
def func2(a:Int,n:Int):Unit={
var sum:Double=0.0;
var prod:Double=1;
var fac:Int=1;
var i:Int=1;
for(i<- 1 to n){
prod = prod*a;
fac= fac*i;
}
}
}

```

```

        sum=sum+prod/fac;
        println(sum)
    }
    print(sum);
}
}

```

27. Write a Scala program to compute the sum of the two given integer values. If the two values are the same, then return triples their sum. (Use anonymous function)

```

object Main{
  def main(args:Array[String]):Unit={
    var result = (a:Int, b:Int)=> if(a==b) {(a+b)*3} else {a+b}
    print(result(3,3))
  }
}

```

28. Write a Scala program to check whether a given positive number is a multiple of 3 or a multiple of 7

```

object Main{
  def main(args:Array[String]):Unit={
    var result = (a:Int, b:Int)=> if(aobject Main{
  def main(args:Array[String]):Unit={
    var i:Int=scala.io.StdIn.readInt();
    if(i%3==0){
      println(i+" is a multiple of 3");
    }
    if(i%7==0){
      println(i+" is a multiple of 7");
    }
  }
}) {(a+b)*3} else {a+b}
    print(result(3,3))
  }
}

```

29. Create an abstract class 'Bank' with an abstract method 'getBalance'. \$100, \$150 and \$200 are deposited in banks A, B and C respectively. 'BankA', 'BankB' and 'BankC' are subclasses of class 'Bank', each having a method named 'getBalance'. Call this method by creating an object of each of the three classes.

```

object Main{
  def main(args:Array[String]):Unit={
    var a = new BankA(500);
    a.getBalance()
  }
}

abstract class Bank{
  var balance:Int =0;

```

```

    def getBalance();
}

class BankA(b:Int) extends Bank{

    this.balance= b;
    def getBalance():Unit={
        println("Balance = "+this.balance);
    }
}

```

30. We have to calculate the area of a rectangle, a square and a circle. Create an abstract class 'Shape' with three abstract methods namely 'RectangleArea' taking two parameters, 'SquareArea' and 'CircleArea' taking one parameter each. The parameters of 'RectangleArea' are its length and breadth, that of 'SquareArea' is its side and that of 'CircleArea' is its radius. Now create another class 'Area' containing all the three methods 'RectangleArea', 'SquareArea' and 'CircleArea' for printing the area of rectangle, square and circle respectively. Create an object of class 'Area' and call all the three methods.

```

object Main{
    def main(args:Array[String]):Unit={
        var a = new Area();
        a.rectangleArea(5,10)
        a.squareArea(10)
        a.circleArea(10)
    }
}

abstract class Shape{
    def rectangleArea(a:Int,b:Int);
    def squareArea(s:Int);
    def circleArea(r:Int);
}

class Area extends Shape{
    def rectangleArea(a:Int,b:Int){
        println(a*b)
    }
    def squareArea(s:Int){
        println(s*s);
    }
    def circleArea(r:Int){
        println(3.14*r*r)
    }
}

```


31. We have to calculate the percentage of marks obtained in three subjects (each out of 100) by student A and in four subjects (each out of 100) by student B. Create an abstract class 'Marks' with an abstract method 'getPercentage'. It is inherited by two other classes 'A' and 'B' each having a method with the same name which returns the percentage of the students. The constructor of student A takes the marks in three subjects as its parameters and the marks in four subjects as its parameters for student B. Create an object of the two classes and print the percentage of marks for both the students.

```
object Main{
  def main(args:Array[String]):Unit={
    var a = new A(93,90,98);
    println(a.getPercentage())
    var b = new B(90,92,94,97);
    println(b.getPercentage());
  }
}

abstract class Marks{
  def getPercentage():Double;
}

class A(s1:Int,s2:Int,s3:Int) extends Marks{
  def getPercentage():Double={
    return (s1+s2+s3)*100.0/300
  }
}

class B(s1:Int,s2:Int,s3:Int,s4:Int) extends Marks{
  def getPercentage():Double={
    return (s1+s2+s3+s4)*100.0/400
  }
}
```

32. (Repeated q29)

GUI Question:

33. Write a program with 4 swing buttons with suitable texts on them. When the user presses a button, a message should appear in the label as to which button was pressed by the user

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

class Buttons extends JFrame implements ActionListener{
  Container c;
  JButton b1,b2,b3,b4;
  JLabel l1;
  Buttons(){
    c = getContentPane();
```

```

        c.setLayout(new FlowLayout());

        b1 = new JButton("Addition");
        b2 = new JButton("Subtraction");
        b3 = new JButton("Multiplication");
        b4 = new JButton("Division");

        l1= new JLabel();
        l1.setFont(new Font("Arial", Font.ITALIC, 32 ));

        c.add(b1);
        c.add(b2);
        c.add(b3);
        c.add(b4);
        c.add(l1);

        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        b4.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e){
        if(e.getSource()==b1){
            l1.setText("You selected Addition");
        }
        else if(e.getSource()==b2){
            l1.setText("You selected Subtraction");
        }
        else if(e.getSource()==b3){
            l1.setText("You selected Multiplication");
        }
        else{
            l1.setText("You selected Division");
        }
    }

    public static void main(String[] args) {
        Buttons q = new Buttons();
        q.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        q.setBounds(300, 300, 700, 350);
        q.setVisible(true);
        q.setTitle("The four buttons");
    }
}

```

34. Write java program to create a registration form using Swing

35. Write a program to search the string in the given program, display number of occurrences of string and replace the string with new string

Search and Replace String

Input String

Enter String to search

No of occurrences

Replace String With

New String

```
import javax.swing.*;

class StringSearcher{
    public static void main(String[] args) {
        // Prompt user for input string
        String input = JOptionPane.showInputDialog("Enter the input
string:");

        // Prompt user for search string
        String search = JOptionPane.showInputDialog("Enter the search
string:");

        // Count the number of occurrences of the search string in the
input string
        int count = 0;
        int index = 0;
        while ((index = input.indexOf(search, index)) != -1) {
            count++;
            index += search.length();
        }

        // Display the number of occurrences
        JOptionPane.showMessageDialog(null, "Number of occurrences: " +
count);

        // Prompt user for the new string
        String replacement = JOptionPane.showInputDialog("Enter the new
string:");

        // Replace all occurrences of the search string with the new string
        String output = input.replaceAll(search, replacement);

        // Display the resulting string
```

```

JOptionPane.showMessageDialog(null, "Resulting string: " + output);
}
}

```

36. Write Java program to display the even numbers and odd numbers from the given list

```

import javax.swing.*;
class Q36{
    public static void main(String[] args) {
        String input = JOptionPane.showInputDialog("Enter numbers
separated by space: ");
        String[] numbers = input.split(" ");
        int a[] = new int[numbers.length];
        int odd=0,even=0;
        for(int i=0;i<numbers.length;i++){
            a[i]= Integer.parseInt(numbers[i]);
            if(a[i]%2 ==0){
                even+=1;
            }
            else{
                odd+=1;
            }
        }
        JOptionPane.showMessageDialog(null, "Number of odd numbers:
"+odd+"\nNumber of even numbers: "+even);
    }
}

```

37. Write a java program using Swing to implement following

```

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

```

```

import java.awt.event.*;

class Counter extends JFrame implements ActionListener{
    Container c;
    JLabel l1,l2;
    JButton inc,dec,res;
    Counter(){
        c=getContentPane();
        c.setLayout(new FlowLayout());

        inc = new JButton("+");
        dec = new JButton("-");
        res = new JButton("RESET");
        l1 = new JLabel("Counter: ");
        l2 = new JLabel("0");
        c.add(l1);
        c.add(l2);
        c.add(inc);
        c.add(dec);
        c.add(res);
        inc.addActionListener(this);
        dec.addActionListener(this);
        res.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e){
        if(e.getSource()==inc){
            int c = Integer.parseInt(l2.getText());
            c+=1;
            l2.setText(""+c);
        }
        else if(e.getSource()==dec){
            int c = Integer.parseInt(l2.getText());
            c-=1;
            l2.setText(""+c);
        }
        else{
            l2.setText("0");
        }
    }
    public static void main(String[] args) {
        Counter c = new Counter();
        c.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        c.setBounds(200,300,750,500);
        c.setVisible(true);
        c.setTitle("Counter");
    }
}

```

38. Write a Swing application called SwingAdder as shown. The "ADD" button adds the

two integers and display the result. The "CLEAR" button shall clear all the text fields.



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

class Adder extends JFrame implements ActionListener{
    Container c;
    JLabel l1, l2, l3;
    JButton add, clr;
    JTextField t1,t2;
    Adder(){
        c=getContentPane();
        FlowLayout l=new FlowLayout();
        l.setAlignment(FlowLayout.CENTER);
        l.setHgap(10000);
        c.setLayout(l);

        l1 = new JLabel("Number 1: ");
        l2 = new JLabel("Number 2: ");
        l3 = new JLabel("Sum: ");

        t1 = new JTextField();
        t1.setPreferredSize(new Dimension(100,20));
        t2 = new JTextField();
        t2.setPreferredSize(new Dimension(100,20));

        add= new JButton("+");
        clr= new JButton("CLEAR");

        c.add(l1);
        c.add(t1);
        c.add(l2);
        c.add(t2);
        c.add(l3);
        c.add(add);
        c.add(clr);

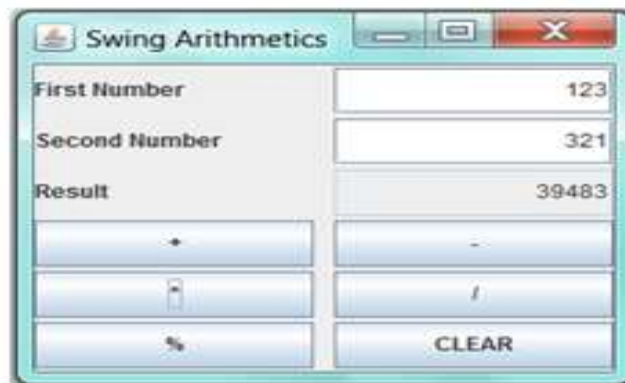
        add.addActionListener(this);
    }
}
```

```

        clr.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e){
        if(e.getSource()==add){
            int num1 = Integer.parseInt(t1.getText());
            int num2 = Integer.parseInt(t2.getText());
            int sum = num1+num2;
            l3.setText("Sum: "+sum);
        }
        else if(e.getSource()==clr){
            t1.setText("0");
            t2.setText("0");
            l3.setText("Sum: ");
        }
    }
    public static void main(String[] args) {
        Adder a = new Adder();
        a.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        a.setBounds(300,400,700,500);
        a.setVisible(true);
        a.setTitle("Adder");
    }
}

```

39. Write java program to Implement the following



```

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

class Calculator extends JFrame implements ActionListener{
    Container c;
    JLabel l1,l2,l3;
    JButton add,sub,mul,div,mod,clr;
    JTextField t1,t2;
    Calculator(){
        c = getContentPane();
        FlowLayout l=new FlowLayout();
    }
}

```

```

l.setHgap(5000);
c.setLayout(l);

l1 = new JLabel("First Number:\t");
t1 = new JTextField("0");
t1.setPreferredSize(new Dimension(100,20));
l2 = new JLabel("Second Number:\t");
t2 = new JTextField("0");
t2.setPreferredSize(new Dimension(100,20));
l3 = new JLabel("Output:\t\t");
add = new JButton("+");
sub = new JButton("-");
mul = new JButton("*");
div = new JButton("/");
mod = new JButton("%");
clr = new JButton("CLEAR");

c.add(l1);
c.add(t1);
c.add(l2);
c.add(t2);
c.add(l3);
c.add(add);
c.add(sub);
c.add(mul);
c.add(div);
c.add(mod);
c.add(clr);
add.addActionListener(this);
sub.addActionListener(this);
mul.addActionListener(this);
div.addActionListener(this);
mod.addActionListener(this);
clr.addActionListener(this);
}

public void actionPerformed(ActionEvent e){
    int num1,num2;
    double out;
    num1 = Integer.parseInt(t1.getText());
    num2 = Integer.parseInt(t2.getText());
    if(e.getSource()==add){
        out=num1+num2;
        l3.setText("Output:\t\t"+out);
    }
    if(e.getSource()==sub){
        out=num1-num2;
        l3.setText("Output:\t\t"+out);
    }
}

```



```

    }
    if(e.getSource()==mul){
        out=num1*num2;
        l3.setText("Output:\t\t"+out);
    }
    if(e.getSource()==div){
        out=(double)(num1)/num2;
        l3.setText("Output:\t\t"+out);
    }
    if(e.getSource()==mod){
        out=num1%num2;
        l3.setText("Output:\t\t"+out);
    }
    if(e.getSource()==clr){
        t1.setText("");
        t2.setText("");
        l3.setText("Output:\t\t");
    }
}

public static void main(String[] args) {
    Calculator c = new Calculator();
    c.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    c.setBounds(300, 400, 500, 300);
    c.setVisible(true);
    c.setTitle("Calculator");
}
}

```