

Program No: 1.

Define a class 'product' with data members pcode, pname and price. Create 3 objects of the class and find the product having lowest price.

```
public class Product {
```

```
    String pcode, pname;
```

```
    float price;
```

```
    public void Setdata (String a, String b, float c)
```

```
    {
```

```
        pcode = a;
```

```
        pname = b;
```

```
        price = c;
```

```
}
```

```
    void display ()
```

```
{
```

```
    System.out.println ("In pcode: " + pcode + " " + pname  
                        + pname + " " + price + " " + price);
```

```
}
```

```
public static void main (String [] args) {
```

```
    product obj1 = new product ();
```

```
    product obj2 = new product ();
```

product obj3 = new product();

obj1. Set data ("P100", "Soap", 50);

obj2. Set data ("P101", "Pen", 10);

obj3. Set data ("P102", "Bodywash", 100);

System.out.println ("The product having lowest price");

if ((obj1.price < obj2.price) && (obj1.price < obj3.price))

{

 obj1.display();

}

else if (obj2.price < obj3.price)

{

 obj2.display();

}

else

{

 obj3.display();

}

}

RESULT.

The program has been successfully executed and output is verified.

OUTPUT.

The product having the lowest price

PCode : 101

Pname : Pen

Price : 10.0

program No: 2.

Add complex numbers.

```
public class comp {
```

```
    int x, y;
```

```
    void get (int a, int b)
```

```
    { x = a; y = b; }
```

```
    void show ()
```

```
{
```

```
    System.out.println ("x + " + "y + " + "i");
```

```
}
```

```
public static void main (String [ ] args) {
```

```
    comp obj1 = new comp();
```

```
    comp obj2 = new comp();
```

```
    comp obj3 = new comp();
```

```
    obj1.get (3, 6);
```

```
    obj2.get (8, 3);
```

```
    obj1.show ();
```

```
    obj2.show ();
```

```
    obj3.x = obj1.x + obj2.x
```

```
    obj3.y = obj1.y + obj2.y
```

obj3.show();

{

}

RESULT.

The program has been executed and output is verified.

OUTPUT.

$$3+6i$$

$$8+3i$$

$$11+9i$$

program No: 3.

Read a matrix from the console and check whether it is symmetric or not.

```
import java.util.*;
public class matrix {
    public static void main(String[] args) {
        Scanner ip = new Scanner(System.in);
        System.out.println("Enter the no of row");
        int row = ip.nextInt();
        System.out.println("Enter the no of column");
        int col = ip.nextInt();
        if (row == col) {
            System.out.println("Matrix is
symmetric");
        } else {
            System.out.println("Matrix is not
symmetric");
        }
    }
}
```

RESULT:

The program has been executed and output is verified.

OUTPUT.

Enter the number of row

5

Enter the number of column

5

Matrix is symmetric.

program No: 4.

Create CPU with attribute price. Create inner class processor (no of cores, manufacturer) and static nested classes RAM (memory, manufacturer). Create an object of CPU and present information of processor and RAM.

public class CPU {

```
    int price;  
    CPU (int p) {  
        this.price = p;  
    }
```

class processor {

```
    int cores;  
    String manufacturer;  
    processor (int n, String m) {  
        this.cores = n;  
        this.manufacturer = m;  
    }
```

void display () {

```
    System.out.println ("No. of cores : " + this.cores);  
    System.out.println ("processor manufacturer " +  
        this.manufacturer);
```

static class RAM &

int memory;

String manufacture;

Ram (int n, String m) {

this. memory = n;

this. manufacture = m;

}

}

void display () {

System.out.println ("price of CPU : " +
this.CPU);

}

Public static void main (String [] args) {

CPU intel = new CPU (23000);

CPU. processor = i-processor = intel. new

processor (4, "intel");

CPU. Ram i-Ram = new Ram (1024, "Asus");

intel. display ();

i-processor. display ();

i-Ram. display ();

}

Output RESULT.

The program has been executed and output
is verified.

OUTPUT.

Price of CPU: 23000

No. of cores: 4

processor manufacturer: Intel

Memory Size: 1024

Memory Manufacturer: ASUS

Program No: 5.

Program to sort string.

Public class SortString {

```
public static void main [String [] args){  
String names [] = { "amal", "Joyeli", "College" };  
String temp;
```

```
int n = names.length
```

```
int i;
```

```
int j;
```

```
for (i=0; i<n; i++)
```

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

```
{
```

```
if ((name[i].compareTo (names[j]) > 0)
```

```
{ temp = names[i];
```

```
names[i] = names[j];
```

```
names[j] = temp;
```

```
} }
```

```
System.out.println ("The sorted array of  
string is : ");
```

```
for (i=0; i<n; i++)
```

```
{
```

```
system.out.println(names[i]);
```

}

}

}

RESULT.

The program has been executed and output is verified.

Output.

The sorted array of String is:

amal

college

Teythi

Program No: 6.

Search an element in an array.

```
import java.util.*;
```

```
public class Searcher {
```

```
    public static void main (String [] args) {
```

```
        int n, i, b, flag = 0;
```

```
        Scanner s = new Scanner (System.in);
```

```
        System.out.println ("Enter number of elements in array");
```

```
        n = s.nextInt();
```

```
        int a[] = new int [n];
```

```
        System.out.println ("Enter " + n + " elements of the array : ");
```

```
        for (i=0; i < n; i++)
```

```
        { a[i] = s.nextInt(); }
```

```
}
```

```
System.out.println ("Enter the element you want to search");
```

```
b = s.nextInt();
```

```
for (i=0; i < n; i++)
```

```
{ if (a[i] == b)
```

```
    flag = 1;  
}  
break; // if element is found  
}  
else  
{ flag = 0; }  
}  
if (flag == 1)  
{  
    System.out.println("Element found at  
position :" + (i+1));  
}  
else  
{  
    System.out.println("Element not found");  
}  
}  
}  
}
```

RESULT.

The program has been executed and the output is verified.

Output :

Enter the number of elements for the array:

5.

Enter the elements of the array:

2

3

4

5

7

Enter the element you want to search:

6.

Element found at position: 4.

Program No: 7.

Perform String Manipulation.

```
Public class Sample_String {  
    public static void main (String [] args) {  
        String str_sample = "superhero";  
        System.out.println ("Length of String: " + str_sample.length());  
        System.out.println ("Character at position 5: " +  
                           str_sample.charAt (5));  
        System.out.println ("Ends with character 'O': " +  
                           str_sample.endsWith ("O"));  
        System.out.println ("Replace 'hero' with  
                           'power': " + str_sample.replace  
                           ("hero", "power");  
    }  
}
```

RESULT

The program has been executed and
Output is verified.

Output:

length of string: 9

character at position 5: h

End with character 'o': true

Replace 'hero' with 'power': Superhero Power

Program No: 8

Program to create a class for employee having attributes eNo, eName, eSalary. Read employee information and search for an employee given eNo, using the concept of objects.

```
import java.util.*;
public class employee {
    int eNo;
    String eName;
    int eSalary;
}

public void read() {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter ID: ");
    eNo = Integer.parseInt(sc.nextLine());
    System.out.print("Enter Name: ");
    eName = sc.nextLine();
    System.out.print("Enter monthly Salary: ");
    eSalary = Integer.parseInt(sc.nextLine());
}
```

```
public void display()
```

{

```
System.out.println("Name: " + empname);
```

}

```
public static void main(String[] args) {
```

```
int i, n = 3;
```

```
int No;
```

```
employee emp[] = new employee[n];
```

```
for (i=0; i<n; i++)
```

```
{ emp[i] = new employee();
```

```
emp[i].read();
```

}

```
System.out.println("Search");
```

```
while (true) {
```

```
Scanner sc = new Scanner(system.in);
```

```
System.out.println("Enter ID: ");
```

```
No = Integer.parseInt(sc.nextLine());
```

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

```
if (emp[i].eno == No) {
```

```
emp[i].display();
```

```
break;
```

}

,

RESULT.

The program has been executed & output is verified.

OUTPUT.

Enter ID: 1001

Enter Name: ashish

Enter monthly Salary: 25000

Enter ID: 101

Enter Name: wilson

Enter monthly Salary: 30000

Enter ID: 102

Enter Name: Ann

Enter monthly Salary: 50000

Search

Enter ID: 101

Name: wilson

*

Program No: 9.

Area of different shapes using overloaded functions.

Public class Shape {

int area (int side)

{ return side * side; }

int area (int l, int b)

{ return l * b; }

double area (double b, double h)

{ return (0.5 * (b * h)); }

double area (double r)

{ return (3.14 * r * r); }

Public static void main (String args) {

Shape obj = new Shape();

System.out.println ("Area of Square: " + obj.area(2));

System.out.println ("Area of rectangle: " + obj.area(4, 4));

System.out.println ("Area of Triangle: " + obj.area(2, 3));

System.out.println ("Area of Circle: " + obj.area(2.5));

}

y

RESULT.

The program has been executed and output is verified.

Output:

Area of square: 1

Area of Rectangle: 8

Area of Triangle: 3.75

Area of Circle: 19.625

Program No: 10.

Create a class 'Employee' with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create class another class 'Teacher' that inherit the properties of class employee and contain its own data members department, Subject taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers.

```
import java.util.*;  
class Employee {  
    int empid;  
    String name, address;  
    double salary;  
    public Employee (int empid, String name,  
                    String address, double salary) {  
        this.empid = empid;  
        this.name = name;
```

-this address = address;

This . Salary = Salary ;

public class Teacher extends Employee

۹

String Subject, department;

```
public Teacher (int empid, String name, String  
address, double salary, String department,  
String subject) {
```

Super (empid, name, address, salary);

this . Subject = Subject;

this . department = department;

۶

void display()

۲

```
System.out.println("Empid:" + this.empid +  
"Name:" + this.name + "Salary:" + this.  
salary + "Address:" + this.address + "department"  
+ this.department + "Subject:" + this.subject);
```

Public static void main (String [] args) {
Scanner sc = new Scanner (System.in);
int n;
System.out.println ("Enter no. of teacher: ");
n = sc.nextInt();
Teacher obj [] = new Teacher [n];
for (int i=0; i<n; i++) {
int j = i+1;
System.out.println ("Enter EmpID of teacher " + j + ":");
int Empid = sc.nextInt();
System.out.println ("Enter Name teacher " + j + ":");
String Name = sc.next();
System.out.println ("Enter Salary of teacher: " + j + ":");
double Salary = sc.nextDouble();
System.out.println ("Enter Address of teacher: " + j + ":");
String Address = sc.next();
System.out.println ("Enter department of teacher: " + j + ":");
String department = sc.next();
System.out.println ("Enter subject of teacher: " + j + ":");
String subjects = sc.next();
obj [i] = new Teacher (Empid, Name, Address,
Salary, department, subjects);
}

```
System.out.println("teacher's list ");  
for (int i=0; i<n; i++)  
{  
    obj[i].display();  
}  
}
```

RESULT.

The program was has been
executed and output is verified.

OUTPUT

Enter number of Teachers:

2.

Enter Empid of teacher 1: 101

Enter Name of teacher 1: John

Enter Salary of teacher 1: 25000

Enter Address of teacher 1: V.P. road

Enter department of teacher 1: Computer Application

Enter Subjects of teacher 1: DBMS

Enter Empid of teacher 2: 102

Enter Name of teacher 2: Don

Enter Salary of teacher 2: 24500

Enter address of teacher 2: J.P. Nagar

Enter department of teacher 2: Computer Application

Enter Subjects of teacher 2: DS

Teacher's List.

Empid: 101 Name: John Salary: 25000 Address: V.P. road
department: Computer Application Subject: DBMS

Empid: 102 Name: Don Salary: 24500 Address: J.P. Nagar
department: Computer Application Subject: DS

Program No:11.

Create a class 'Person' with data members Name, gender, Address, Age and constructor to initialize the data members and another class employee that inherits the properties of class person and also contains its own data members like Empid, Company-name, qualification, & salary and its own constructor. Create another 'teacher'-that inherits the properties of class employee and contains its own data members like Subject, Department, Teacherid and also contains constructors and methods to display the data members. Use array of objects of display details of N teachers.

```
import java.util.*;
```

```
class person
```

```
{
```

```
    String name, gender, address;  
    int age, int empid, String Cname, String Qualification;  
    double Salary, String Subject;
```

public person (String name, String gender,
String address, int age) {
super();

this.name = name;

this.gender = gender;

this.address = address;

this.age = age;

,

}

Class Employee extends Person {

int empId;

String cname, qualification;

double salary;

public Employee (String name, String gender,
String address, int age, int empId,
String cname) {

super(name, gender, address, age);

this.empId = empId;

this.cname = cname;

this.qualification = qualification;

this.salary = salary;

,

}

class Teacher extends employee
{ String Subject, department;
int teacherid;

public Teacher (String name, String gender, String address,
int age, int empid, String cname,
qualification, salary);

this. Subject = subject;

this. department = department;

this. teacherid = teacherid;

*

void display()

{

System.out.println (" Personal details ");

System.out.println (" Name : " + this.name + " Gender : " +
this.gender + " Age : " + this.age);

System.out.println (" Employee details ");

System.out.println (" EmpId : " + this.empid + "
Company-name : " + this.cname + " Salary : "
+ this.salary + " Address : " + this.address + "
qualification : " + this.qualification);

System.out.println (" Teacher details ");

System.out.println (" TeacherId : " + this.teacherid + "
department : " + this.department + " Subject "
+ this.subject);

```
public class Main {
```

```
    public static void main (String [] args) {
```

```
        Scanner s = new Scanner (System.in);
```

```
        int n;
```

```
        System.out.println ("Enter number of Teachers:");
```

```
        n = s.nextInt();
```

```
        Teacher obj[] = new Teacher [n];
```

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

```
            System.out.print ("Enter the person name:");
```

```
            String name = s.next();
```

```
            System.out.print ("Enter the Gender:");
```

```
            String gen = s.next();
```

```
            System.out.print ("Enter the Address:");
```

```
            String add = s.next();
```

```
            System.out.print ("Enter Age:");
```

```
            int age = s.nextInt();
```

```
            System.out.print ("Enter EmployeeId:");
```

```
            int id = s.nextInt();
```

```
            System.out.print ("Enter Company name:");
```

```
            String cname = s.next();
```

```
            System.out.print ("Enter the Salary:");
```

```
            double sal = s.nextDouble();
```

```
            System.out.print ("Enter Qualification:");
```

```
            String qual = s.next();
```

```
System.out.println("Enter teacher id");
int tid = s.nextInt();
System.out.println("Enter Department");
String dept = s.next();
System.out.println("Enter the Subject");
String sub = s.next();
```

```
obj[i] = new Teacher(name, gen, ade, age,
id, cname, qual, sal, sub, dept, tid);
```

```

}
for (int i=0; i<n; i++)
    obj[i].display();
}
}
```

RESULT.

The program was has been executed and the output is verified.

Output:

Person details

Name: John Gender: male Age: 23

Employer details

EmpId: 1001 Company-name: XYZ, Salary: 24000
Address: VPC road, Qualification: B.E

Teacher details

Name: Teacher Ed: 2001, department: CS
subject: DBMS



Program No: 12

Write a program has class publisher, Book, Literature, Fiction. Read the information and print the details of books from either the category, using inheritance.

```
import java.util.*;
```

```
class publisher {  
    String pubname;  
    publisher()  
}
```

```
Scanner s = new Scanner (System.in);  
System.out.println ("Enter publisher name");  
pubname = s.next();
```

```
}
```

```
class Book extends Publisher  
{
```

```
String title, author;  
int price;
```

Book()

```
{ Scanner s = new Scanner(System.in);  
System.out.println("Enter Title of book");  
title = s.next();  
System.out.println("Enter Author's name");  
author = s.next();  
System.out.println("Enter Price");  
price = s.nextInt();  
}
```

y

class Literature extends Book

{

Literature()

{

```
System.out.println("Literature Books");  
}
```

void display()

{

```
System.out.println("Publisher name: " + pubname);  
System.out.println("Title of Book: " + title);  
System.out.println("Author name: " + author);  
System.out.println("Price: " + price);  
}
```

}

class Fiction extends Literature

{ Fiction()

{

System.out.println("Fictional Books");

void display()

{

super.display();

}

public static void main(String args[])

{ int n;

Scanner s = new Scanner(System.in);

int a = s.nextInt();

Literature L[] = new Literature[a];

for(int i=0; i<a; i++)

{ L[i] = new Literature(); }

System.out.println("Enter no. of fiction books");

int b = nextInt();

Fiction F[] = new Fiction[b];

for(int i=0; i<b; i++)

{ F[i] = new Fiction(); }

int no;

System.out.println("Enter your choice");

```
no = s.nextInt();
```

```
int type = no;
```

```
switch (no)
```

```
{
```

```
case 1:
```

```
System.out.println ("Details of literature book");
```

```
for (int i = 0; i < a; i++)
```

```
    w[i].display();
```

```
break;
```

```
case 2:
```

```
System.out.println ("Details of fiction book");
```

```
for (int i = 0; i < b; i++)
```

```
    f[i].display();
```

```
break;
```

```
default:
```

```
System.out.println ("Wrong Input");
```

```
}
```

```
4
```

```
}
```

RESULT

The program has been successfully executed and output is verified.

Output.

1. literature Books

2. Fictional Books

Enter your choice of Books:

1

Details of literature Books

Publisher name: abc

Title of Book: Pride and Prejudice

Author's name: Jane Austen

price: 400.

Publisher name: abc

Title of the book: The Great Gatsby

Author name: Fitzgerald.

price: 580.

Program No:13

Create classes Student and sports. Create another class Result inherited from student and sports. Display the academic and sports score of student.

class Student

{ int maths = 85;

int science = 72;

int English = 88;

int SocialScience = 70;

}

class Sports extends Student

String sport = "Football";

int goal = 2;

int assist = 1;

int grace = 20°;

}

public class Result extends Sports {

public void display ()

{

System.out.println(" Academic Result");

System.out.println("Maths: " + this.maths + "\nScience: " + this.science + " English: " + this.english + "\nSocial Science: " + this.social_science);

System.out.println("Sports Results");

System.out.println("In Sport: " + this.sport + "\nGoal: " + this.goal + " Assist: " + this.assist + "\nGrace: " + this.grace);
}

public static void main(String[] args)

{

Result obj = new Result();

obj.display();

}

RESULT

The program has been successfully executed and output is verified.

Output.

Academic Result

Maths : 85

Science : 72

English : 88

Social Science : 70

Sports Result

Sport : Football

Goal : 2

Assist : 1

Grace : 20