

LAB MANUAL

OBJECT ORINTED

PROGRAMMING USING JAVA

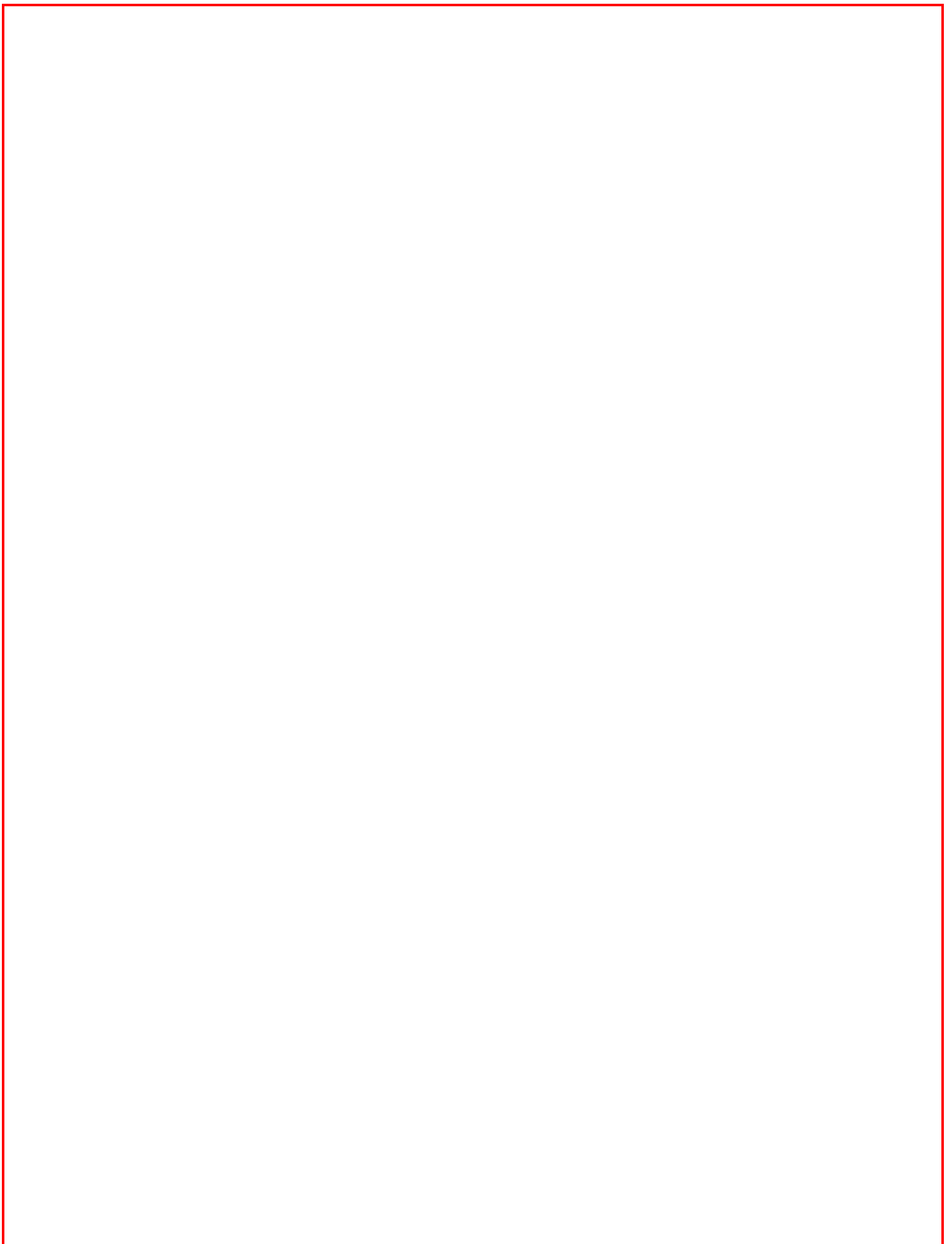
IIInd Year

*Department of Computer
Engineering RAJA JAIT SINGH
GOVT.POLYTECNIC SEC-72
NEEMKA FARIDABAD.*



Prepared by Gaytri Tanwar

Department of computer science





List of Practical

S. No	Experiment
1	Consider we have a Class of Cars. Use this class to create another class Company that tracks the models it creates.
2	Create the Employee base class and also some behaviors properties that all employee must have for that Company.
3	Suppose the Airport personals want to maintain records. Create a class Airport that has data like name, id, and address.
4	Create a whole menu driven hospital management system using concepts of OOP like classes, inheritance.
5	Create a class called Musicians to contains three methods string, wind, and perc. Each of these methods should initialize a string array.
6	Write three derived classes inheriting functionality of base class person and with added unique features of students and employees and functionality to assign.

7	Using the concept of multiple inheritance create classes; shape; circle; square; cube; sphere; cylinder. Your classes may only have the class variable.
8	Write a program to create class Person. Write the class definition, the constructors, set methods, get methods and for all classes.
9	Old MacDonald had a farm and several types of animals. Define new class for this and implement animals.
10	Write a program with student as abstract class and create derive classes engineering, Medicine and science students.

PRACTICAL-1

1. Consider we have a Class of Cars under which Santro Xing, Alto and Wagon R represents individual Objects. In this context each Car Object will have its own, model, Year of Manufacturing, Color, Top speed, etc. which form Properties of the Car class and the associated actions i.e., object functions like Create (), Sold (), display () form the Methods of Cars Class. Use this class to create another class Company that tracks the models it creates.

Program

```
Class Car
{
    String model;
    int yearofmfg;
    String color;
    int topspeed;
    Int qtysold; void Create(String m1,int
y1,String c1,int tsp)
{
    model=m1;
    yearofmfg=y1;
    color=c1;
    topspeed=tsp;
}
void Sold(int s1)
{ qtysold=s1;
}
void Display()
{
    System.out.println("Model : "+model);
    System.out.println("Year of Manufacturing : "+model);
    System.out.println("Color : "+color);
    System.out.println("Top Speed : "+topspeed);
    System.out.println("Qty Sold : "+qtysold);
}
}

Public class Company {

    publicstaticvoid main(String[] args) {
        // TODO Auto-generated method stub
        Car santro=new Car();
        Car alto=new Car();
    }
}
```

```
Car wagonr=new Car();
santro.Create("Santro
Xing",2019,"Red",100); santro.Sold(500);
santro.Display();
alto.Create("Alto",2018,"White",98);
alto.Sold(750);
alto.Display();
wagonr.Create("Wagon R",2017,"Red",98);
wagonr.Sold(400);
wagonr.Display();
```

```
}
```

```
}
```

Output: -

Model : Santro Xing

Year of Manufacturing : Santo Xing

Color : Red

Top Speed : 100

Qty sold : 500

Model : Alto

Year of Manufacturing: Alto

Color: White

Top Speed: 98

Qty sold: 525

Model: Wagon R

Year of Manufacturing: Wagon R

Color: Red

Top Speed: 100

Qty sold: 480

PRACTICAL: - 2

2. In a software company Software Engineers Sr. Software Engineers, Module Lead, Technical Lead, Project Lead, Project Manager, Program Manager, Directors all are the employees of the company but their work, perks, role, responsibilities differ. Create the Employee base class would provide the common behaviors of all types of employees and also some behaviors properties that all employees must have for that company. Also include search method to search an employee by name.

Program

```
import java.util.*;
class Employee
{
    String name;
    int id;
}
Class TypeEmployee extends Employee
{
    String work;
    int perks;
    String roles;
    String resp;
    void setData(String sname, int sid, String swork, int sperks, String sroles, String sresp)
    {
        name=sname;
        id=sid;
        work=swork;
        perks=sperks;
        roles=sroles;
        resp=sresp;
    }
}
```



```

void displayDetails()
{
System.out.println("Name of Employee : " +name);
System.out.println("Id of Employee : " +id);
System.out.println("Work of Employee : " +work);
System.out.println("Perks of Employee : " +perks);
System.out.println("Role of Employee : " +role);
System.out.println("Responsibility of Employee : " +resp);
}
String getName()
{
    return(name);
}
}
publicclass SoftwareCompany {
publicstaticvoid main(String ar[ ] )
{
    TypeEmployee emp[ ]=new TypeEmployee[8];
    int i,f=0;
    String sname;
    for(i=0;i<8;i++)
    {
        emp[i]= new TypeEmployee ();
    }
    emp[0].setData("Rohit",1,"Software Testing",50000,"Software Engineer","Software Testing &
Maintenance");
    emp[1].setData("Rajesh",2,"Software Development",75000,"Software Engineer","Software
Designing");
    emp[2].setData("Harsh",3" Technical Advisor",60000,"Module Lead","Support and operations");
    emp[3].setData("Vinod",4" Technical Asstt",60000,"Technical Lead","Support and operations");
    emp[4].setData("Amarjeet",5,"Project Planning",78000,"Project Lead", Project Supervisor");
    emp[5].setData("Chandan",6,"Project Advisor",70000,"Project Manager", Project
Development");
    emp[6].setData("Sachin",7,"Programmer",70000,"Program Manager",Development and
operations");
    emp[7].setData("Vandana",8," Overall. Operations",95000,"Director", Management
Control");
    for(i=0;i<8;i++)

```

```
{  
    System.out.println("Employee details");  
    emp[i].displayDetails();  
    System.out .println(" _____ ");  
}
```

```
System.out.println("Enter name of Employee to Search");  
Scanner sc =new Scanner(System.in);
```

```

sname=sc.next();
for(i=0;i<8;i++)
{
    if(emp[i].getName().equals(sname))
    {
        System.out.println("Employee is searched");
        emp[i].displayDetails ();
        f=1;
    }
}
if(f==0)
{
    System.out.println("Employee us not Found");
}
}
}

```

Output: -

Employee Details
 Name Of Employee: Rohit
 Id of Employee: 1
 Works of Employee: Software Testing
 Perks of Employee: 50000
 Role of Employee: Software Engineers
 Responsibility of Employee: Software Testing & Maintenance

Employee Details
 Name Of Employee: Rajesh
 Id of Employee: 1
 Works of Employee: Software Development
 Perks of Employee: 75000
 Role of Employee: Software Engineers
 Responsibility of Employee: Software Designing

Employee Details

Name Of Employee: Harsh

Id of Employee: 1

Works of Employee: Technical Advisor

Perks of Employee: 60000

Role of Employee: Module Lead

Responsibility of Employee: Supports and operations

PRACTICAL: - 3

3. Suppose the airport personals want to maintain records for the arrival and departures of the planes. Create a class Airport that has data like name, id, and address. Create twoore classes for Arrival and departure implementing Airport that will have track of planes(their name ,id , arrival, time or departure time and a counter to count the number of arrivals and also include the necessary methods to access the information.

Program

```
import java..util.*;
//Airport class
class Airport
{
    String name;
    int id;
    void get1()
    {
        Scanner sc =new Scanner(System.in);
        System.out.println(:Enter Airport Name);
        name=sc.nextLine();
        System.out.println("Enter Airport id");
        id=integer.parseInt(sc.nextLine());
        System.out.println("Enter Address/City");
        address=sc.nextLine();
    }
    void show()
    {
        System.out.println("Airport Name:"+name);
        System.out.println("Airport id:"+id);
        System.out.println("Address/city:"+address);
    }
}
//Plane arrival class
class Arrival extends Airport
{
    String pname;
    int pid;
    String arrival_time;
    void get_arrival()
    {
```

```
    get1();//calling function of airport directly
Scanner sc= new scanner (System.in);
System.out.println("Enter Palne Name");
pname =sc.nextLine();
System.out.println("Enter Plane Id");
pid=Integer.parseInt(sc.nextLine());
System.out.printl("Enter Arrival time:");
arrival time=sc.next();
}
```

```

void show arrival ()
{
    System.out.println("plane Name:"+pname);
    System.out.println("Plane id:"+pid);
    System.out.println("Arrival time:"+arrival time);
}
}
// Passenger class
class Passenger extends Airport
{
    String psname;
    String padd;
    void vet passenger ()
    {
        Scanner sc=new Scanner (System.in);
        System.out.println("Enter passenger name:");
        psname=sc.nextaLine();
        System.out.println("Enter passenger address:");
        padd=sc.nextLine();
    }
    void show passenger()
    {
        System.out.println("Passenger Name:"+psname);
        System.out.println("Address:"+padd);
    }
    String get psname ()
    {
        return psname;
    }
}
//Departure class
class Departure extends Airport
{
    String pname;
    int pid;
    String dept time;
    void get departure ()
    {

```

```
get();//calling function of airport directly
Scanner sc= new scanner (System.in);
System.out.println("Enter Plane name to departure");
pname=sc.nextLine();
System.out.println("Enter Plane id to departure");
pid=Integer.parseInt(sc.nextLine());
System.out.println("Enter Departure time");
dept time=sc.nextLine();
```

```
}
```



```

void show arrival ()
{
    System.out.println("Plane Name:"+pname);
    System.out.println("Plane id:"+pid);
    System.out.println("Arrival time:"+dept time");
}
}
//main program
publicclass Airportpersonals{
    publicstaticvoid main(String[ ] args) {
//TODO Auto-generated method sub
//arrival planes details
    Arrival are[ ]= new Arrival [8];
    int i =0,j,f=0;
    String sname,ch;
    Scanner sc=new scanner (System.in);
for(i=0;i<8;i++)
    {
//allocating memory
arr[i] = new Arrival ();
}
i=0;
do
{
// accepting the details
    System.out.println("Enter Details of Arrival plane"+(i+1));
    arr[i].get arrival ();
    System.kut.println("Press y for another details");
    ch=sc.nextline();
    i++;
}
while(ch.equallgnoreCase("Y"));
//showing the details
    for(j=0;j<i;j++)
// accepting the details
        System.out.println("Enter details of departure plane"+(i+1));
        dep[i].get departure ();
        System.out.println("Press Y for another details");

```

```
        ch=sc.next();
    }
    while(ch.equalsIgnoreCase("Y"));
//showing the details
for(j=0;j<i;j++)
{
    System.out.println("Enter details of departure plane"+(i+1));
}
//Passenger details
```

```

Passenger pas[ ]= new passenger [8];
i=0;
for(i=0;i<8;i++)
{
//allocating memory
arr[i] = new Arrival ();
}
i=0;
do
{
// accepting the details
    System.out.println("Enter details of departure plane"+(i+1));
    dep[i].get departure ();
    System.out.println("Press Y for another details");
    ch=sc.next();
}
while(ch.equalsIgnoreCase("Y"));
//showing the details
for(j=0;j<i;j++)
{
    System.out.println("Enter details of departure plane"+(i+1));
}
    System.out.println("Enter the name of passenger to search");
    sname=sc.nextLine();
for(j=0;j<i;j++)
{
    if(pas[j].get psname (). equals (sname))
    {
        System.out.println("passenger is searched");
        f=1;
    }
}
}

```

Output: -

Enter Details of Arrival plane 1

Enter Airport Name

Indira Gandhi

Enter Airport id

1001

Enter Address/City

New Delhi

Enter plane name

Spice Air

Enter plain id

20001

Enter Arrival time

4:10 am

Press y for another details

Details of arrival plain 1

Plane name

Spice Air

Plain id

20001

Total number of arrivals

1

Enter Details of Departure Plane 1

Enter Airport Name

Shivaji

Enter Airport id

1001

PRACTICAL:- 4

4. Create a whole menu driven hospital management system using concept of OOP like classes, inheritance, Include information about the following:

a. Patient-name, registration id, age, disease, etc.

b. Staff-id, name, designation, salaries.

Program

```
package first;
import jav.util.*;
class general
{
String name, id;
Scanner sc= new Scanner(System.in);
void show data()
{
    System.out.printl("\nid:"+id);
    System.out.printl("\nName:"+name);
}
}
class staff extends general
{
String desg;
int salary;
void new staff()
{
get data();
System.out.println("designation");
    desg=sc.nextline();
    System.out.println("salary");
Salary=sc.nextlnt();
}
```

```

}
void staff info()
{
show data();
System.out.println("\nDesignation:"+desg);
System.out.println("\nsalary:"+salary);
}
}
class patient extends general
{
String diseases, sex;
int age;
void new patient()
{
get data();
System.out.println("disease");
    disease =sc.nextLine();
System.out.println("sex");
    sex =sc.nextLine();
System.out.println("age");
    age =sc.nextLine();
}
void patient info()
{
Show data();
System.out.println("disease: "+disease);
System.out.println("sex:"+sex);
System.out.println("age:"+age);
}
}
class hos{
public static void main([String[] args){
    //TODO Auto-generated method sub
int ch;
Scanner inp=new Scanner(System.in);
System.out.println("Enter 1 for Patient Data");
System.out.println("Enter 2 for Staff Data");
System.out.println("Your choice");
}
}

```

```
if (ch==1)
{
    patient pt= new patient();
    pt.new patient();
    pt.patient info();
}
(if ch==2)
{
    staff st=new staff();
```



```
st.new staff();
st.staff info();
}
if(ch>=3)
    System.out.println("Wrong Choice");
}
```

Output:-

Enter 1 for patient Data

Enter 2 for staff Data

Your choice

1

Id:-101

Name:-Ram

Diseases:-Fever

Sex:- Male

Age:-20

Enter 1 for Patient Data

Enter 2 for Staff Data

Your choice

2

Id:-1010

Name:- Dr. Rakesh

Designation:- Sr. Doctor

Salary;- 75000

Id:-1010

Name:- Dr. Rakesh

Designation:- Sr. Doctor
Salary: 75000

Enter 1 for Patient Data
Enter 2 for Staff Data

Your choice
3 wrong choice

PRACTICAL:- 5

5. Create a class called Musicians to contain three methods string (), wind () and perc (). Each of these methods should initialize a string array to contain the following instruments:

- veena,guitar, sitar, sarod and mandolin under string ()
- Flute, clarinet saxophone, nadhaswaram and piccolo under wind ()
- Tabla, mridangam, bangos, drums and tambour under perc ()

It should also display the contents of the arrays that are initialized. Create a derived class called TypeInsto contain a method called get () and show (). The get () method must display a means as follows.

Type of instruments to be displayed:

- a. String instruments
- b. wind instruments
- c. Percussion instruments

The show () method should display the relevant detail according to our choice. The base class variables must be accessible only to its derived classes.

Program

package first;

```
import java.util.*;
class Musicians
{
    protected String sting[]=new String[5];
    protected String win[]=new String[5];
    protected String per[]=new String[5];
    void string()
    {
        sting[0]="Veena";
        sting[1]="guitar";
        sting[2]="sitar";
        sting[3]="sarod";
        sting[4]="mando";
    }
    void wind()
    {
        win[0]="flute";
        win[1]="clarinet";
        win[2]="saxophone";
        win[3]="nadhaswaram";
        win[4]="piccolo";
    }
    void perc()
    {
        per[0]="tabla";
        per[1]="mridangam";
        per[2]="bangos";
        per[3]="drums"
        per[4]="tambour";
    }
    void show_string()
```

```

        {
            for(int i=0;i<5;i++)
            {
                System.out.println(sting[i]);
            }
        }
void show_wind()
{
    for(int i=0;i<5;i++)
    {
        System.out.println(win[i]);
    }
}
void show_perc()
{
    for(int i=0;i<5;i++)
    {
        System.out.println(per[i]);
    }
}
}
class typeInsto extends Musicians
{
int ch;
void get()
{
    System.out.println("1 String instruments");
    System.out.println("2 Wind instruments");
    System.out.println("3 Percussions instruments");
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter your choice");
    ch=sc.nesxtInt();
}
void show()
{
    if(ch==1)
    {
        string();
        show_string();
    }
    if(ch==2)
    {
        wind();

```

```
        show_wind();
    }
    if(ch==3)
    {
        perc();
        show_perc();
    }
    if(ch>=4)
        System.out.println("Wrong Choice");
}
```

```
}  
public class musician {  
  
    public static void main(String[] args) {  
        //TODO Auto-generated method stub  
        typeInsto t1=new typeInsto();  
        t1.get();  
        t1.show();  
    }  
}
```

Output :-

```
1 String instruments  
2 Wind instruments  
3 Percussions instruments  
Enter your choice  
1  
Veena  
guitar  
sitar  
sarod  
mando  
1 String instruments  
2 Wind instruments  
3 Percussions instruments  
Enter your choice  
2  
flute  
clarinet  
saxophone  
nadhaswaram  
piccolo  
1 String instruments  
2 Wind instruments  
3 Percussions instruments  
Enter your choice  
3  
tabla
```

mridangam

bangos

drums

tambour

1 String instruments

2 Wind instruments

3 Percussions instruments

Enter your choice

4

Wrong Choice

PRACTICAL: - 6

6. Write three derived classes inheriting functionality of base class person (should have a member function that ask to enter name and age) and with added unique features of student, and employee, and functionality to assign, change and delete records of student and employee.

Program

```
import java.util.Scanner;
class person
{
String name;
int age;
Scanner sc=new Scanner(System.in);
Void get_data()
{
    System.out.print("Name : ");
    name = sc.nextLine();

    System.out.print("Age : ");
    age = sc.nextInt();
}
void show_data()
{
    System.out.print("\nName : "+name);
    System.out.print("\nAge : "+age);
}
}
class student extends person
{
int m,marks;
void new_student()
{
get_data();
System.out.print("Roll No. : ");
rn = sc.nextInt();

    System.out.print("Marks : ");
    Marks = sc.nextInt();
}
void student_info()
```

```
{
show_data();
System.out.print("\nRollno : "+rn);
System.out.print("\nMarks : "+marks);
}
}
class employees extends person
{
String dept;
int pay;
void new_employee()
```

```

{
get_data();
System.out.print("Department : ");
    dept = sc.next();
    System.out.print("Pay : ");
    pay = sc.nextInt();
}
void emp_info()
{
show_data();
System.out.println("\nDepartment : "+dept);
System.out.println("\nPay : "+pay);
}
}
class studemp {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
int ch;
Scanner inp=new Scanner(System.in);
System.out.println("Enter 1 for Student Data");
System.out.println("Enter 2 for Employee Data");
System.out.println("Your Choice");
ch=inp.nextInt();
if(ch==1)
{
    Student st=new student();
    st.new_student();
    st.student_info();
}
if(ch==2)
{
    employee e1=new employees();
    e1.new_employee();
    e1.emp_info();
}
if(ch>=3)
    System.out.println("Wrong Choice");
}
}

```

Output :-

Enter 1 for Student Data

Enter 2 for Employee Data

Your Choice

1

Name : Ramesh

Age : 18

Rollno : 10251

Marks : 255

Enter 1 for Student Data

Enter 2 for Employee Data

Your Choice

2

Name : sachin

Age : 25

Department : sales

Pay : 24000

PRACTICAL :- 7

7. Using the concept of multiple inheritance create classes: Shape, Circle, Square, Cube, Sphere, Cylinder. Your classes may only have the class variable specified in the table below and the methods Area and/or Volume to output their area and/or volume.

Class	Class Variable	Constructor	Base class
Shape	String name	Shape()	
Circle	double radius	Circle(double r, String n)	Shape
Square	double side	Square(double s, String n)	Shape
Cylinder	double height	Cylinder(double h, double r, String n)	Circle
Sphere	None	Sphere(double r, String n)	Circle
Cube	None	Cube(double s, String n)	Square

Program

```
//Base Class Shape
Class Shape
{
    String name;
    double ar;
    Shape()
    {
        name="";
        ar=0.0;
    }
    void show()
    {
        System.out.print("Shape"+name);
    }
}
//Circle Class
class Circle extends Shape
{
    double r;
    Circle(double r1,String n)
    {
        r=r1;
        name=n;
    }
}
```

```

    }
    void area()
    {
        ar=3.14*r*r;
        System.out.print("Area of Circle : "+ar);
    }
}
//Square class
class Square extends Shape
{
    double s;
    Square(double s1,String n)
    {
        s=s1;
        name=n;
    }
    void area()
    {
        ar=s*s;
        System.out.println("Area of Square : "+ar);
    }
}
}
Cylinder Class
class Cylinder extends Circle
{
    double h;
    Cylinder(double h1,double r1,String n1)
    {
        super(r1,n1);
        h=h1;
    }
    void area()
    {
        ar=(2*3.14*r*h)+(2*3.14*(r*r));
        System.out.println("Area of Cylinder : "+ar);
    }
}
}
//Sphere class
class Sphere extends Circle
{
    Sphere(double r1,String n1)
    {
        super(r1,n1);
    }
}
}

```

```
name=n1;
}
void area()
{
    ar=(4*3.14*(r*r));
    System.out.println("Area of Sphere : "+ar);
}
}
//cube class
class Cube extends Square
```



```

{
    Cube(double s1,String n1)
    {
        super(s1,n1);
    }
    void area()
    {
        ar=(6*(s*s));
        System.out.println("Area of cube : "+ar);
    }
}
//main program
publicclass diff_area {
publicstaticvoid main(String ar[])
{
    Circle c1=new Circle(2.5,"Circle");
    c1.area();
    Square s1=new Square(3.6,"Square");
    s1.area();
    Cylinder c2=new Cylinder(2.6,3.0,"Cylinder");
    c2.area();
    Sphere s2=new Sphere(4.5,"Sphere");
    s2.area();
    Cube c3=new Cube(4.3,"Cube");
    c3.area();
}
}

```

Output

```

Area of Circle : 19.625
Area of Square : 12.96
Area of Cylinder : 105.504
Area of Sphere : 254.34
Area of Cube : 110.94

```

PRACTICAL:- 8

8. Write a program to create class Person.

- a. Make two classes, Student and Instructor, inherit from Person. A person has a name and year of birth.**
- b. A student has a major, student id.**
- c. An instructor has a salary, subject .**

Write the class definitions, the constructors, set methods, get methods and for all classes.

Program

```

import java.util.Scanner;
class Person
{
String name;
int birthyear;
Scanner sc=new Scanner(System.in);
void get_data()
{
    System.out.print("Name : ");
    String n=sc.nextLine();
    setName(n);
    System.out.print("Year of Birth : ");
    int y = sc.nextInt();
    setBirthYear(y);
}
void show_data()
{
    System.out.println("\nName : "+getName());
    System.out.println("\nAge : "+getBirthYear());
}
void setName(String n1)
{
    name=n1;
}
void setBirthYear(int y1)
{
    birthyear=y1;
}
String getName()
{
    return name;
}
int getBirthYear()
{
    return birthyear;
}
}
class student extends Person
{
int id;
String major;

```

```
void new_student()
```

```
{
```

```
  get_data();
```

```
  System.out.print("ID : ");
```

```
  int i = sc.nextInt();
```

```
  setId(i);
```

```
  System.out.print("Major : ");
```

```
  String m= sc.next();
```

```

setMajor(m);
}

void student_info()
{
show_data();
System.out.println("\nID : "+getId());
System.out.println("\nMajor : "+getMajor());
}
void setMajor(String m1)
{
    major=m1;
}
void setId(int i1)
{
    id=i1;
}
String getMajor()
{
    return major;
}
int getId()
{
    return id;
}
}

//instructor class
class instructor extends Person
{

int salary;
String subject;
void new_instructor()

{

get_data();
System.out.print("Salary : ");
int s =sc.nextInt();
setSalary(s);
}
}

```

```
System.out.print("Subject : ");  
String sub= sc.next();  
setSubject(sub);  
}
```

```
Void instructor_info()
```

```
{  
show_data();  
System.out.println("\nSalary : "+getSalary());  
}
```

```

System.out.println("\nSubject : "+getSubject());
}
void setSubject(String s1)
{
    subject=s1;
}
void setSalary(int s1)
{
    salary=s1;
}
String getSubject()
{
    return subject;
}
int getSalary()
{
    return salary;
}
}

class getset {

    Public static void main(String[] args) {
        // TODO Auto-generated method stub

    Int ch;
    Scanner inp=new Scanner(System.in);
    System.out.println("Enter 1 for Student Data");
    System.out.println("Enter 2 for Instructor Data");
    System.out.println("Your Choice");
    ch=inp.nextInt();
    if(ch==1)
    {
        Student st=new student();
        st.new_student();
        st.student_info();
    }
    if (ch==2)
    {
        instructor e1=new instructor();
        e1.new_instructor();
        e1.instructor_info();
    }
}

```

```
if(ch>=3)
    System.out.println("Wrong Choice");
}
```

Output:

Enter 1 for Student Data
Enter 2 for Instructor Data

Your Choice

1

Name : Ravi

Year of Birth : 1999

ID : 101

Major : Computer

Name : Ravi

Age : 1999

ID : 101

Major : Computer

Enter 1 for Student Data

Enter 2 for Instructor Data

Your Choice

2

Name : Rohit

Year of Birth : 1987

Salary : 25000

Subject : Electronics

Name : Rohit

Age : 1987

Salary : 25000

Subject : Electronics

PRACTICAL :- 9

9. Old MacDonald had a farm and several types of animals. Every animal shared certain characteristics: they had a type (such as cow, chick or pig) and each made a sound (moo, cluck or oink). An Interface defines those things required to be an animal on the farm. Define new classes for the Old MacDonald that implement the Animal and Farm class. Create an array of objects of animals to define the different types of animal in the farm. Also create appropriate methods to get and set the properties.

Program

```
package first;
Interface Animal
{
    public      String gettypes();
    public      void settype(String x);
    public      String getsound();
    public      void setsound(String x);
}
```

```

class cow implements Animal
{
    String type,sound;
    public void settype(String x)
    {
        type=x;
    }
    public String gettype()
    {
        return type;
    }
    public void setsound(String x)
    {
        sound=x;
    }
    Public String getsound()
    {
        return sound;
    }
}
class chick implements Animal
{
    String type,sound;
    public void settype(String x)
    {
        type=x;
    }
    public String gettype()
    {
        return type;
    }
    public void setsound(String x)
    {
        sound=x;
    }
    Public String getsound()
    {
        return sound;
    }
}
public class macdonald {

    public static void main(String[] args) {

```

```
//TODO Auto-generated method stub
Animal c[]=new Animal[2];
c[0]=new cow();
c[0].settype("cow");
c[0].setsound("moo");
c[1]=new chick();
c[1].settype("chick");
c[1].setsound("cluck");
for(int i=0;i<2;i++)
{
```

```
System.out.println("Animal name "+(i+1)+" : "+c[i].gettype());  
System.out.println("Animal sound : "+c[i].getsound());
```

```
}
```

```
}
```

```
}
```

Output :-

Animal name 1: cow

Animal sound: moo

Animal name 2 : chick

Animal sound: cluck

PRACTICAL :- 10

- 10. Write a program with Student as abstract class and create derived classes Engineering, Medicine and Science from base class Student. Create the objects of the derived classes and process them and access them using an array of pointers of type base class Student.**

Program

```
abstract class Students
{
    private String name;
    private String stream;
    public Students(String nm, String st) {
        name=nm;
        stream =st;
    }
    //abstract method
    abstract void marks();

    @Override
    public String toString() {
        return "Name="+name+"::Stream="+stream;
    }
}
class Engineering extends Students {

    private int mark;

    Engineering(String nm String st, int m) {
        super(nm, st);
        mark=m;
    }

    @Override
    void marks() {
```

```

        System.out.println("Marks : "+mark);
    }
}
class Medicine extends Students {
    private int mark;

    Medicine(String nm, String st, int m) {
        super(nm,st);
        mark=m;
    }
    @Override
    void marks() {
        System.out.println("Marks : "+mark);
    }
}
class Science extends Students {
    private int mark;

    Science(String nm, String st, int m) {
        super(nm, st);
        mark=m;
    }
    @Override
    void marks() {
        System.out.println("Marks : "+mark);
    }
}
public class stuengmedsc {
    public static void main(String ar[])
    {
        Students s[]=new Students[3];
        s[0]=new Engineering("Ravi","Engineering",525);
        s[1]=new Medicine("Ramesh","Medicine",252);
        s[2]=new Science("Rohit","Science",512);

        for(int i=0;i<3;i++)
        {
            }
        }
    }
}

```

Output :-

```
S      ut.println(s[i].toString());s[i].marks();
y      }
s
t
e
m
.
o
```

Name=Ravi::Stream=Engineering

Marks : 525

Name=Ramesh::Stream=Medicine

Marks : 252

Name=Rohit::Stream=Science

Marks : 512