

## DAY 4

1. ;create multilevel inheritance for

```
//Vehicle
//Four_wheeler
//Petrol_Four_Wheeler
//FiveSeater_Petrol_Four_Wheeler
//Baleno_FiveSeater_Petrol_Four_Wheeler
package day_4;

class Vehicle
{
    void type(String type) {
        System.out.println("This is a "+type);
    }
}

class Four_wheeler extends Vehicle
{
    void wheels(int wheels) {
        System.out.println("It has "+wheels+" wheels");
    }
}

class Petrol_Four_wheeler extends Four_wheeler
{
    void fuelType(String fuel) {
        System.out.println("It runs on "+fuel);
    }
}

class FiveSeater_Petrol_Four_wheeler extends Petrol_Four_wheeler
{

```

```

        void seatingCapacity(int capacity) {
            System.out.println("It is a "+capacity+"-seater");
        }
    }

    class Baleno_FiveSeater_Petrol_Four_wheeler extends FiveSeater_Petrol_Four_wheeler
    {
        void model(String model) {
            System.out.println("Model : "+model);
        }
    }

    public class multilevel_vehicle {

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

            Baleno_FiveSeater_Petrol_Four_wheeler car = new
            Baleno_FiveSeater_Petrol_Four_wheeler();

            car.model("Baleno");

            car.type("car");

            car.wheels(4);

            car.fuelType("petrol");

            car.seatingCapacity(5);

        }
    }

```

Output

Model : Baleno

This is a car

It has 4 wheels

It runs on petrol

It is a 5-seater

2. Demonstrate the use of the super keyword

```
package day_4;
```

```
class Parent
```

```
{
```

```
    String name ="parent class";
```

```
    Parent(){
```

```
        System.out.println("Parent Constructor");
```

```
    }
```

```
    void show() {
```

```
        System.out.println("parent method");
```

```
    }
```

```
}
```

```
class Child extends Parent
```

```
{
```

```
    String name ="child class";
```

```
    Child(){
```

```
        super();
```

```
        System.out.println("Child constructor");
```

```
    }
```

```
    void display() {
```

```
        super.show();
```

```
        System.out.println("Parent name :"+super.name);
```

```
    }
```

```
}
```

```
public class Super {
```

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

```
        // TODO Auto-generated method stub
```

```
        Child c = new Child();
```

```
        c.display();
```

```
    }
```

```
}
```

Output

Parent Constructor

Child constructor

parent method

Parent name :parent class

Create Hospital super class and access this class inside the patient child class and access properties from Hospital class.

```
package day_4;
```

```
class Hospitals
```

```
{
```

```
    void hospInfo(String name) {
```

```
        System.out.println("Welcome to "+name+" hospital !");
```

```
    }
```

```
}
```

```
class Doctor extends Hospitals
```

```
{
```

```
    void doctorInfo(String name , String spec) {
```

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

```
        System.out.println("Specialization: " + spec);
```

```
    }
```

```
}
```

```
class Gynac extends Doctor
```

```
{
```

```
    void gynacDuty(String dutyTime) {
```

```
        System.out.println("Gynaecologist Duty Time: " + dutyTime);
```

```
    }
```

```
}
```

```
class Endo extends Doctor
```

```
{
```

```
    void endoDuty(String dutyTime) {
```

```
        System.out.println("Endocrinologist Duty Time: " + dutyTime);
```

```
    }
```

```
}
```

```
class Cardiac extends Doctor
```

```
{
```

```
    void cardiacDuty(String dutyTime) {
```

```
        System.out.println("Cardiologist Duty Time: " + dutyTime);
```

```
    }
```

```
}
```

```
class Operation extends Cardiac
```

```
{
```

```
    void surgerySchedule(String time) {
```

```
        System.out.println("Scheduled Operation Time: " + time);
```

```
    }
```

```
}
```

```
class OPD extends Cardiac
```

```
{  
    void opdTime(String time) {  
        System.out.println("OPD Time: " + time);  
    }  
}
```

```
class Nurse extends Hospitals
```

```
{  
    void nurseInfo(String name, String shift) {  
        System.out.println("Nurse Name: " + name);  
        System.out.println("Shift: " + shift);  
    }  
}
```

```
class Accountant extends Hospitals
```

```
{  
    void accountantInfo(String name) {  
        System.out.println("Accountant: " + name);  
    }  
}
```

```
class Payments extends Accountant
```

```
{  
    void paymentDetails(double amount) {  
        System.out.println("Payment Amount: Rs." + amount);  
    }  
}
```

```
class Documentation extends Accountant
```

```

{
    void docStatus(String status) {
        System.out.println("Documentation Status: " + status);
    }
}

```

```

public class Hospital_hierar {

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        System.out.println("*****Hospital Info*****");

        Gynac g = new Gynac();
        g.hospInfo("CityCare");
        g.doctorInfo("Dr. Neha", "Gynaecologist");
        g.gynacDuty("9AM - 3PM");

        System.out.println("\n*****Cardiology Operation*****");
        Operation o = new Operation();
        o.hospInfo("CityCare");
        o.doctorInfo("Dr. Raj", "Cardiologist");
        o.cardiacDuty("10AM - 4PM");
        o.surgerySchedule("1PM");

        System.out.println("\n*****OPD*****");
        OPD opd = new OPD();
        opd.hospInfo("CityCare");
        opd.doctorInfo("Dr. Ajay", "Cardiologist");
        opd.opdTime("11AM - 2PM");
    }
}

```

```

        System.out.println("\n*****Nurse*****");

        Nurse n = new Nurse();

        n.hospInfo("CityCare");

        n.nurseInfo("Sister Latha", "Morning");


        System.out.println("\n*****Account Department*****");

        Payments p = new Payments();

        p.hospInfo("CityCare");

        p.accountantInfo("Mr. Kiran");

        p.paymentDetails(3500.50);


        Documentation d = new Documentation();

        d.hospInfo("CityCare");

        d.accountantInfo("Mr. Kiran");

        d.docStatus("Completed");

    }

}

```

## Output

```

*****Account Department*****

Welcome to CityCare hospital !

Accountant: Mr. Kiran

Payment Amount: Rs.3500.5

Welcome to CityCare hospital !

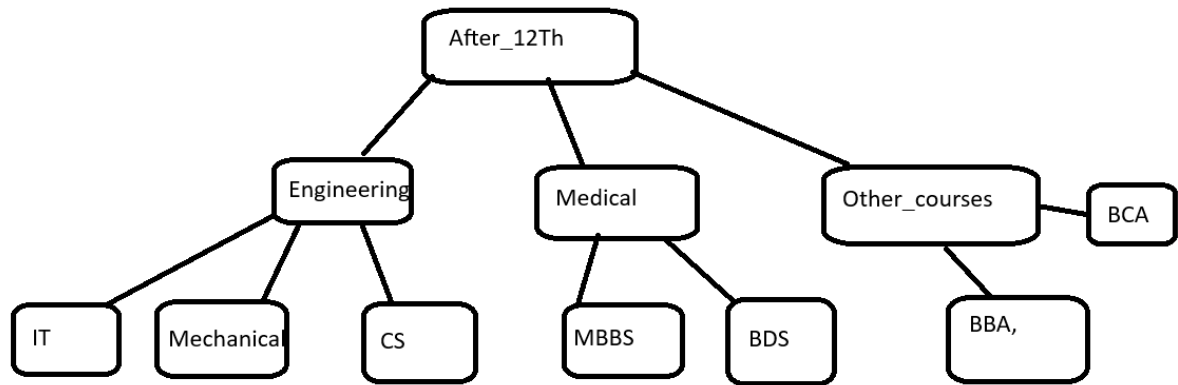
Accountant: Mr. Kiran

Documentation Status: Completed

```



### 3. Create Hierarchical inheritance



```
package day_4;
```

```
class After_12th{
```

```
    void category() {
```

```
        System.out.println("Options after 12 th");
```

```
    }
```

```
}
```

```
class Engineering extends After_12th {
```

```
    void engineering() {
```

```
        System.out.println("You selected Engineering stream.");
```

```
    }
```

```
}
```

```
class IT extends Engineering {
```

```
    void branchIT() {
```

```
        System.out.println("Welcome to IT Branch.");
```

```
    }
```

```
}
```

```
class Mechanical extends Engineering {
```

```
    void branchMechanical() {
```

```
        System.out.println("Welcome to Mechanical Branch.");
```

```
}  
}
```

```
class CS extends Engineering {  
    void branchCS() {  
        System.out.println("Welcome to Computer Science Branch.");  
    }  
}
```

```
class Medical extends After_12th {  
    void medical() {  
        System.out.println("You selected Medical stream.");  
    }  
}
```

```
class MBBS extends Medical {  
    void branchMBBS() {  
        System.out.println("Welcome to MBBS.");  
    }  
}
```

```
class BDS extends Medical {  
    void branchBDS() {  
        System.out.println("Welcome to BDS.");  
    }  
}
```

```
class Other_Courses extends After_12th {  
    void others() {  
        System.out.println("You selected Other Courses.");  
    }  
}
```

```
}  
}
```

```
class BBA extends Other_Courses {  
    void courseBBA() {  
        System.out.println("Welcome to BBA Course.");  
    }  
}
```

```
class BCA extends Other_Courses {  
    void courseBCA() {  
        System.out.println("Welcome to BCA Course.");  
    }  
}
```

```
public class course {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        System.out.println("--- IT Branch ---");  
  
        IT it = new IT();  
        it.category();  
        it.engineering();  
        it.branchIT();  
  
        System.out.println("\n--- Mechanical Branch ---");  
        Mechanical mech = new Mechanical();  
        mech.category();  
        mech.engineering();  
        mech.branchMechanical();  
    }  
}
```

```
System.out.println("\n--- CS Branch ---");
```

```
CS cs = new CS();
```

```
cs.category();
```

```
cs.engineering();
```

```
cs.branchCS();
```

```
System.out.println("\n--- MBBS Branch ---");
```

```
MBBS mbbs = new MBBS();
```

```
mbbs.category();
```

```
mbbs.medical();
```

```
mbbs.branchMBBS();
```

```
System.out.println("\n--- BDS Branch ---");
```

```
BDS bds = new BDS();
```

```
bds.category();
```

```
bds.medical();
```

```
bds.branchBDS();
```

```
System.out.println("\n--- BBA Course ---");
```

```
BBA bba = new BBA();
```

```
bba.category();
```

```
bba.others();
```

```
bba.courseBBA();
```

```
System.out.println("\n--- BCA Course ---");
```

```
BCA bca = new BCA();
```

```
bca.category();
```

```
bca.others();
```

```
bca.courseBCA();
```

}

}

### Output

Options after 12 th

You selected Other Courses.

Welcome to BBA Course.

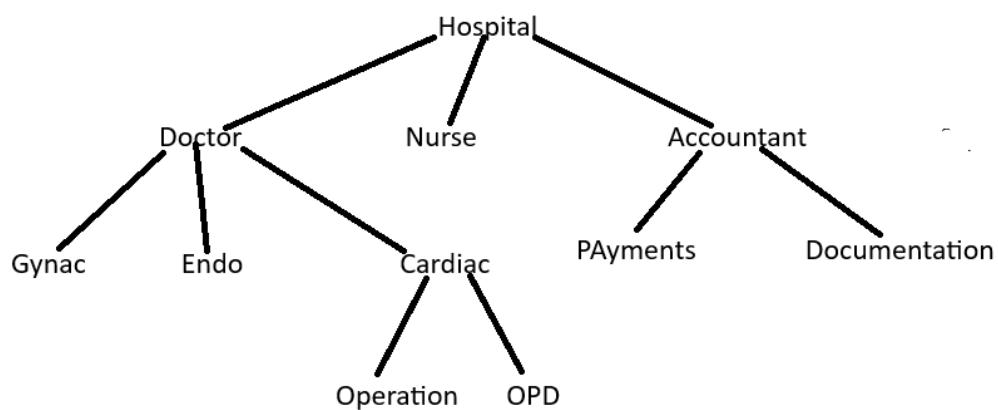
--- BCA Course ---

Options after 12 th

You selected Other Courses.

Welcome to BCA Course.

### 4. Create practice on this



```
package day_4;
```

```
class Hospitals
```

```
{
```

```
    void hospInfo(String name) {
```

```
        System.out.println("Welcome to "+name+" hospital !");
```

```
    }
```

```
}
```

```
class Doctor extends Hospitals
```

```
{  
    void doctorInfo(String name , String spec) {  
        System.out.println("Doctor Name: " + name);  
        System.out.println("Specialization: " + spec);  
    }  
}
```

```
class Gynac extends Doctor
```

```
{  
    void gynacDuty(String dutyTime) {  
        System.out.println("Gynaecologist Duty Time: " + dutyTime);  
    }  
}
```

```
class Endo extends Doctor
```

```
{  
    void endoDuty(String dutyTime) {  
        System.out.println("Endocrinologist Duty Time: " + dutyTime);  
    }  
}
```

```
class Cardiac extends Doctor
```

```
{  
    void cardiacDuty(String dutyTime) {  
        System.out.println("Cardiologist Duty Time: " + dutyTime);  
    }  
}
```

```
class Operation extends Cardiac
```

```
{  
    void surgerySchedule(String time) {  
        System.out.println("Scheduled Operation Time: " + time);  
    }  
}
```

class OPD extends Cardiac

```
{  
    void opdTime(String time) {  
        System.out.println("OPD Time: " + time);  
    }  
}
```

class Nurse extends Hospitals

```
{  
    void nurseInfo(String name, String shift) {  
        System.out.println("Nurse Name: " + name);  
        System.out.println("Shift: " + shift);  
    }  
}
```

class Accountant extends Hospitals

```
{  
    void accountantInfo(String name) {  
        System.out.println("Accountant: " + name);  
    }  
}
```

class Payments extends Accountant

```
{
```

```
        void paymentDetails(double amount) {  
            System.out.println("Payment Amount: Rs." + amount);  
        }  
    }  
}
```

```
class Documentation extends Accountant  
{  
    void docStatus(String status) {  
        System.out.println("Documentation Status: " + status);  
    }  
}
```

```
public class Hospital_hierar {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
  
        System.out.println("*****Hospital Info*****");  
  
        Gynac g = new Gynac();  
        g.hospInfo("CityCare");  
        g.doctorInfo("Dr. Neha", "Gynaecologist");  
        g.gynacDuty("9AM - 3PM");  
  
        System.out.println("\n*****Cardiology Operation*****");  
  
        Operation o = new Operation();  
        o.hospInfo("CityCare");  
        o.doctorInfo("Dr. Raj", "Cardiologist");  
        o.cardiacDuty("10AM - 4PM");  
        o.surgerySchedule("1PM");  
    }  
}
```



```

System.out.println("\n*****OPD*****");

OPD opd = new OPD();

opd.hospInfo("CityCare");

opd.doctorInfo("Dr. Ajay", "Cardiologist");

opd.opdTime("11AM - 2PM");


System.out.println("\n*****Nurse*****");

Nurse n = new Nurse();

n.hospInfo("CityCare");

n.nurseInfo("Sister Latha", "Morning");

System.out.println("\n*****Account Department*****");

Payments p = new Payments();

p.hospInfo("CityCare");

p.accountantInfo("Mr. Kiran");

p.paymentDetails(3500.50);

Documentation d = new Documentation();

d.hospInfo("CityCare");

d.accountantInfo("Mr. Kiran");

d.docStatus("Completed");

    }
}

```

### Output

\*\*\*\*\*Account Department\*\*\*\*\*

Welcome to CityCare hospital !

Accountant: Mr. Kiran

Payment Amount: Rs.3500.5

Welcome to CityCare hospital !

Accountant: Mr. Kiran

Documentation Status: Completed

## Polymorphism

1. Create a class Calculator with the following overloaded add()

1.add(int a, int b)

2.add(int a, int b, int c)

3.add(double a, double b)

```
package day_4_polymorphism;
```

```
public class Calculator {  
    void add(int a, int b) {  
        System.out.println("Sum :"+(a+b));  
    }  
    void add(int a, int b, int c) {  
        System.out.println("Sum :"+(a+b+c));  
    }  
    void add(double a, double b) {  
        System.out.println("Sum :"+(a+b));  
    }  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        Calculator c = new Calculator();  
        c.add(1, 2);  
        c.add(1, 2, 3);  
        c.add(3.0, 5.0);  
    }  
}
```

## Output

Sum :3

Sum :6

Sum :8.0

2. Create a base class Shape with a method area() that prints a message.  
Then create two subclasses                      Circleàoverride area() to calculator and  
print area of circle                              Rectangleà override area() to calculate  
and print area of a rectangle

```
package day_4_polymorphism;
```

```
class Shape
{
    void area() {
        System.out.println("Area calculation");
    }
}
```

```
class Circle extends Shape
{
    void area() {
        double radius = 5.0;
        double area = 3.14 * radius * radius;
        System.out.println("Area of Circle: " + area);
    }
}
```

```
class Rectangle extends Shape
{
    void area() {
        int length = 10;
        int breadth = 6;
        int area = length * breadth;
        System.out.println("Area of Rectangle: " + area);
    }
}
```

```

}

public class Shapes {

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        Circle c = new Circle();

        c.area();

        Rectangle r = new Rectangle();

        r.area();

    }

}

```

Output:

Area of Circle: 78.5

Area of Rectangle: 60

3. Create a Bank class with a method `getInterestRate()`  
subclasses: SBI → return 6.7%  
ICICI → return 7.0%                      HDFC → return 7.5%
- package day\_4\_polymorphism;

create

```

class Bank

{

    double getInterestRate(){

        return 0.0;

    }

}

```

```

class SBI extends Bank

```

```
{  
    double getInterestRate() {  
        return 6.7;  
    }  
}
```

class ICICI extends Bank

```
{  
    double getInterestRate() {  
        return 7.0;  
    }  
}
```

class HDFC extends Bank

```
{  
    double getInterestRate() {  
        return 7.5;  
    }  
}
```

public class Bank\_interest {

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

```
        // TODO Auto-generated method stub
```

```
        SBI sbi = new SBI();
```

```
        ICICI icici = new ICICI();
```

```
        HDFC hdfc = new HDFC();
```

```
        System.out.println("SBI Interest Rate: " + sbi.getInterestRate() + "%");
```

```
        System.out.println("ICICI Interest Rate: " + icici.getInterestRate() + "%");
```

```
System.out.println("HDFC Interest Rate: " + hdfc.getInterestRate() + "%");
```

```
}
```

```
}
```

#### Output:

SBI Interest Rate: 6.7%

ICICI Interest Rate: 7.0%

HDFC Interest Rate: 7.5%

#### 4. Runtime Polymorphism with constructor Chaining

create a class vehicle with a constructor that prints “Vehicle Created”

Create a subclass Bike that override a method and uses super() in constructor

package day\_4\_polymorphism;

```
class Vehicle{  
    Vehicle(){  
        System.out.println("Vehicle Created");  
    }  
    void start() {  
        System.out.println("Vehicle Started");  
    }  
}
```

```
class Bike extends Vehicle{  
    Bike(){  
        super();  
        System.out.println("Bike created");  
    }  
    void start() {  
        System.out.println("Bike is starting");  
    }  
}
```

```

    }
}

public class Vehicle_demo {

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        Bike b = new Bike();

        b.start();

    }

}

```

### Output

Vehicle Created

Bike created

Bike is starting

Create an abstract class SmartDevice with methods like turnOn(), turnOff(), and performFunction().

Create child classes:

- SmartPhone: performs calling and browsing.
- SmartWatch: tracks fitness and time.
- SmartSpeaker: plays music and responds to voice commands.
- 
- Write code to store all objects in an array and use polymorphism to invoke their performFunction().

```
package day_4_other;
```

```

abstract class SmartDevice {

    abstract void turnOn();

    abstract void turnOff();
}

```

```
        abstract void performFunction();  
    }  
}
```

```
class SmartPhone extends SmartDevice {  
    void turnOn() {  
        System.out.println("SmartPhone is turning on");  
    }  
    void turnOff() {  
        System.out.println("SmartPhone is turning off");  
    }  
    void performFunction() {  
        System.out.println("SmartPhone: Making calls and browsing the internet");  
    }  
}
```

```
class SmartWatch extends SmartDevice {  
    void turnOn() {  
        System.out.println("SmartWatch is turning on");  
    }  
    void turnOff() {  
        System.out.println("SmartWatch is turning off");  
    }  
    void performFunction() {  
        System.out.println("SmartWatch: Tracking fitness and showing time");  
    }  
}
```

```
class SmartSpeaker extends SmartDevice {  
    void turnOn() {  
        System.out.println("SmartSpeaker is turning on");  
    }  
}
```



```

    }

    void turnOff() {

        System.out.println("SmartSpeaker is turning off");

    }

    void performFunction() {

        System.out.println("SmartSpeaker: Playing music and responding to voice
commands");

    }

}

```

```

public class Smart_Device_demo {

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        SmartDevice[] devices = new SmartDevice[3];

        devices[0]=new SmartPhone();

        devices[1]=new SmartWatch();

        devices[2]=new SmartSpeaker();

        for(SmartDevice d : devices) {

            d.performFunction();

        }

    }

}

```

#### Output:

SmartPhone: Making calls and browsing the internet

SmartWatch: Tracking fitness and showing time

**2.**Design an interface Bank with methods deposit(), withdraw(), and getBalance(). Implement this in SavingsAccount and CurrentAccount classes.

- Use inheritance to create a base Account class.
  - Demonstrate method overriding with customized logic for withdrawal (e.g., minimum balance in SavingsAccount).
- 

### 3

Create a base class Vehicle with method start().

Derive Car, Bike, and Truck from it and override the start() method.

- Create a static method that accepts Vehicle type and calls start().
- Pass different vehicle objects to test polymorphism.

```
package day_4_other;
```

```
class Vehicle{  
    void start() {  
        System.out.println("Vehicle startingg..");  
    }  
}
```

```
class Car extends Vehicle{  
    void start() {  
        System.out.println("Car startingg..");  
    }  
}
```

```
class Bike extends Vehicle {  
    void start() {  
        System.out.println("Bike starting...");  
    }  
}
```

```

    }
}

class Truck extends Vehicle {

    void start() {

        System.out.println("Truck starting...");

    }

}

public class Vehicle_demo {

    public static void startVehicle(Vehicle v) {

        v.start();

    }

    public static void main(String[] args) {

        Vehicle myCar = new Car();

        Vehicle myBike = new Bike();

        Vehicle myTruck = new Truck();

        startVehicle(myCar);

        startVehicle(myBike);

        startVehicle(myTruck);

    }

}

```

**Output:**

Car startingg..

Bike starting...

Truck starting...

---

#### 4.

Design an abstract class Person with fields like name, age, and abstract method getRoleInfo().

Create subclasses:

- Student: has course and roll number.
- Professor: has subject and salary.
- TeachingAssistant: extends Student and implements getRoleInfo() in a hybrid way.
- Create and print info for all roles using overridden getRoleInfo().

```
package day_4_other;
```

```
abstract class Person {  
    String name;  
    int age;  
  
    abstract void getRoleInfo();  
}
```

```
class Student extends Person {  
    String course;  
    int roll_no;  
  
    void getRoleInfo() {  
        System.out.println("Student Info:");  
        System.out.println("Name: " + name + ", Age: " + age);  
        System.out.println("Course: " + course + ", Roll No: " + roll_no);  
    }  
}
```

```
class Professor extends Person {  
  
    String subject;  
  
    double salary;  
  
    void getRoleInfo() {  
        System.out.println("Professor Info:");  
  
        System.out.println("Name: " + name + ", Age: " + age);  
  
        System.out.println("Subject: " + subject + ", Salary: ₹" + salary);  
    }  
}
```

```
class TeachingAssistant extends Student {  
  
    String assignedProfessor;  
  
    void getRoleInfo() {  
        System.out.println("Teaching Assistant Info:");  
  
        System.out.println("Name: " + name + ", Age: " + age);  
  
        System.out.println("Course: " + course + ", Roll No: " + roll_no);  
  
        System.out.println("Assisting Professor: " + assignedProfessor);  
    }  
}
```

```
public class Person_info {  
  
    public static void main(String[] args) {  
  
        Student s = new Student();  
  
        s.name = "Ravi";  
  
        s.age = 20;  
  
        s.course = "B.Tech CSE";  
  
        s.roll_no = 101;
```

```

Professor p = new Professor();

p.name = "Dr. Meera";

p.age = 45;

p.subject = "Data Structures";

p.salary = 95000;


TeachingAssistant ta = new TeachingAssistant();

ta.name = "Anjali";

ta.age = 22;

ta.course = "M.Tech AI";

ta.roll_no = 202;

ta.assignedProfessor = "Dr. Meera";

Person[] people = {s, p, ta};


for (Person person : people) {

    person.getRoleInfo();

    System.out.println("-----");

}

}

}

```

#### OUTPUT:

Name: Dr. Meera, Age: 45

Subject: Data Structures, Salary: ₹95000.0

-----

Teaching Assistant Info:

Name: Anjali, Age: 22

Course: M.Tech AI, Roll No: 202

Assisting Professor: Dr. Meera

-----

