**LAB:4**

**/\*Question 1: Vehicle Inheritance**

**Scenario:**

**Create a parent class Vehicle with fields like brand, speed, and a method displayInfo().**

**Create child classes:**

**Car – with an additional field numDoors**

**Bike – with an additional field type (e.g., "sports", "cruiser")**

**Task:**

**Use inheritance to reuse Vehicle properties.**

**Create objects for Car and Bike, set their values, and display all the details.\*/**

class Vehicle{

// fields

String brand;

int speed;

void displayInfo(){

System.out.println("Brand : " +brand);

System.out.println("Speed : " +speed+"km/hr");

}

}

class Car extends Vehicle{

//Additional field

int numDoors;

//Display Car Information

void displayCarInfo(){

displayInfo(); //call parent method

System.out.println("Number of Doors : " +numDoors);

}

}

class Bike extends Vehicle{

//Additional field

String type;

//Display Bike info

void displayBikeInfo(){

displayInfo(); //call parent method

System.out.println("Type : "+type);

}

}

class VehicleTest{

public static void main(String args[]){

// create car object

Car car = new Car();

car.brand = "Tata";

car.speed = 150;

car.numDoors = 4;

System.out.println("Car Details : ");

car.displayCarInfo();

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

// Create object for bike

Bike bike = new Bike();

bike.brand = "Yamaha";

bike.speed = 120;

bike.type = "Sports";

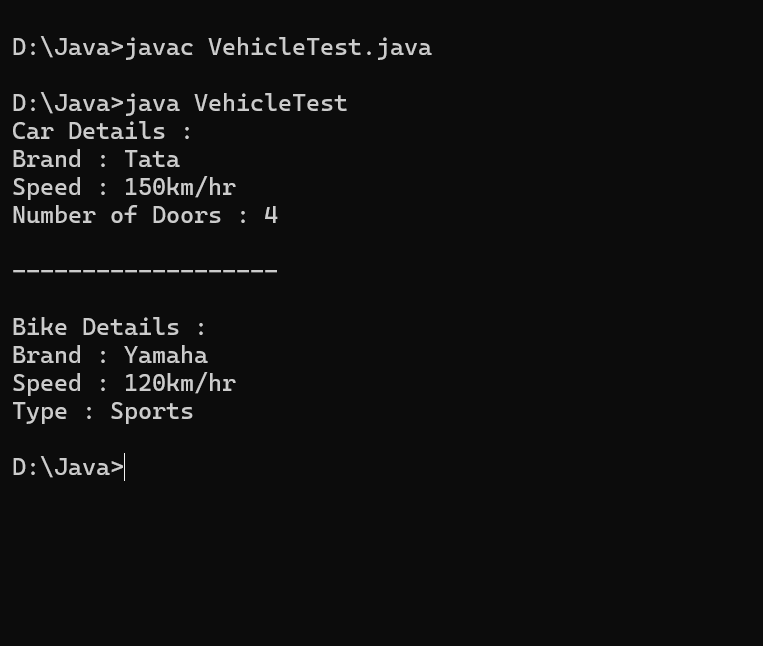
System.out.println("Bike Details : ");

bike.displayBikeInfo();

}

}

**OUTPUT:**



**/\*Question 2: Employee Management System**

**Scenario:**

**Create a base class Employee with fields name, id, and salary.**

**Create subclasses:**

**Manager – with a field department**

**Developer – with a field programmingLanguage**

**Task:**

**Use constructors to initialize values.**

**Write a method showDetails() in the parent and override it in each subclass to show specific details.\*/**

class Employee{

// Fields

String name;

int id;

double salary;

//Constructor

Employee(String name,int id,double salary){

this.name = name;

this.id = id;

this.salary = salary;

}

//Method to show details

void showDetails(){

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

System.out.println("Employee ID : " +id);

System.out.println("Salary : "+salary);

}

}

class Manager extends Employee{

//additional Field

String department;

//constructor

Manager(String name,int id,double salary,String department){

super(name,id,salary);

this.department = department;

}

//overriding method

void showDetails(){

super.showDetails();

System.out.println("Department : "+department);

}

}

class Developer extends Employee{

//additional fields

String programmingLanguage;

//constructor

Developer(String name,int id,double salary,String programmingLanguage){

super(name,id,salary);

this.programmingLanguage = programmingLanguage;

}

//overriding

void showDetails(){

super.showDetails();

System.out.println("Programming Language : " +programmingLanguage);

}

}

//Main class

class EmployeeTest{

public static void main(String args[]){

//create manager object

Manager mgr = new Manager("Sowbarnika", 101,75000,"HR");

System.out.println("Manager Details:");

mgr.showDetails();

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

// create object for developer

Developer dev = new Developer("Rahul",103,85000,"Java");

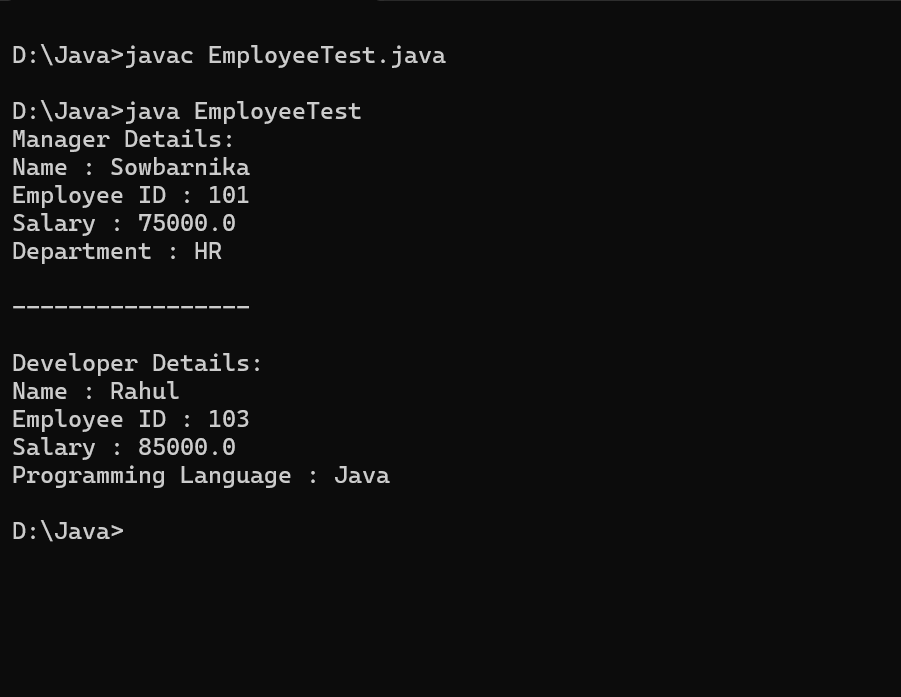
System.out.println("Developer Details:");

dev.showDetails();

}

}

**OUTPUT :**



**/\*Question 3: College System – Hierarchical Inheritance**

**Scenario:**

**Create a base class Person with fields like name and age.**

**Create 2 subclasses:**

**Student – with fields course and rollNo**

**Professor – with fields subject and employeeId**

**Task:**

**Demonstrate how both Student and Professor inherit from Person.\*/**

class Person{

// Fields

String name;

int age;

//Constructor

Person(String name,int age){

this.name = name;

this.age = age;

}

//Display method

void display(){

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

System.out.println("Age: "+age);

}

}

class Student extends Person{

// Fields of student

String course;

int rollNo;

//Constructor

Student(String name,int age,String course,int rollNo){

super(name,age);

this.course = course;

this.rollNo = rollNo;

}

//display method

void display(){

super.display(); //call base class method

System.out.println("Course : " +course);

System.out.println("Roll Number : "+rollNo);

}

}

class Professor extends Person{

//Fields of Professor

String subject;

int employeeId;

//Constructor

Professor(String name,int age,String subject,int employeeId){

super(name,age);

this.subject = subject;

this.employeeId = employeeId;

}

//display method

void display(){

super.display();

System.out.println("Subject : "+subject);

System.out.println("Employee ID : "+employeeId);

}

}

class CollegeSystem{

public static void main(String args[]){

Student student = new Student("Sowbarnika",20,"Information Technology",645);

Professor professor = new Professor("Dr.Radhi",45,"Mathematics",756);

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

student.display();

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

professor.display();

}

}

**//OUTPUT :**

