**ICS1312 – JAVA PROGRAMMING LABORATORY**

**DATE : 30.7.2025**

**ASSIGNMENT : 2**

**TITLE : Inheritence and polymorphism**

**ROLLL NO : 3122247001017**

**LEARNING OBJECTIVE:**

* **To work with interface in java**
* **To create abstract class in java**
* **To use super keyword to call parent class**

A paper with text and images

AI-generated content may be incorrect.

CLASS DIAGRAM :

A paper with writing on it

AI-generated content may be incorrect.

CODE:

import java.util.Scanner;

class MotorVehicle {

    String brand;

    double maxSpeed;

    String colour;

    String vehicleNumber;

    public MotorVehicle(String brand, double maxSpeed, String colour, String vehicleNumber) {

        this.brand = brand;

        this.maxSpeed = maxSpeed;

        this.colour = colour;

        this.vehicleNumber = vehicleNumber;

    }

    void displayInfo() {

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

        System.out.println("Max Speed: " + maxSpeed + " km/h");

        System.out.println("Colour: " + colour);

        System.out.println("Vehicle Number: " + vehicleNumber);

    }

}

class Car extends MotorVehicle {

    String fuelType;

    public Car(String brand, double maxSpeed, String colour, String vehicleNumber, String fuelType) {

        super(brand, maxSpeed, colour, vehicleNumber);

        this.fuelType = fuelType;

    }

    void displayInfo() {

        System.out.println("\nCar Details:");

        super.displayInfo();

        System.out.println("Fuel Type: " + fuelType);

    }

    void displayInfo(boolean displayNumber) {

        if (displayNumber) {

            System.out.println("Vehicle Number: " + vehicleNumber);

        } else {

            System.out.println("Vehicle Number not displayed.");

        }

    }

}

class ElectricCar extends Car {

    int batteryCapacity;

    int chargeRemaining;

    public ElectricCar(String brand, double maxSpeed, String colour, String vehicleNumber, String fuelType,

                       int batteryCapacity, int chargeRemaining) {

        super(brand, maxSpeed, colour, vehicleNumber, fuelType);

        this.batteryCapacity = batteryCapacity;

        this.chargeRemaining = chargeRemaining;

    }

    void displayInfo() {

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

        super.displayInfo();

        System.out.println("Battery Capacity: " + batteryCapacity + " kWh");

        System.out.println("Charge Remaining: " + chargeRemaining + "%");

    }

    void displayInfo(boolean displayNumber, int chargeRemaining) {

        if (displayNumber) {

            System.out.println("Vehicle Number: " + vehicleNumber);

        }

        System.out.println("Charge Remaining: " + chargeRemaining + "%");

    }

}

public class Driver{

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter MotorVehicle Details:");

        System.out.print("Brand: ");

        String mvBrand = sc.nextLine();

        System.out.print("Max Speed: ");

        double mvSpeed = sc.nextDouble();

        sc.nextLine();

        System.out.print("Colour: ");

        String mvColour = sc.nextLine();

        System.out.print("Vehicle Number: ");

        String mvNumber = sc.nextLine();

        MotorVehicle mv = new MotorVehicle(mvBrand, mvSpeed, mvColour, mvNumber);

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

        System.out.print("Brand: ");

        String carBrand = sc.nextLine();

        System.out.print("Max Speed: ");

        double carSpeed = sc.nextDouble();

        sc.nextLine();

        System.out.print("Colour: ");

        String carColour = sc.nextLine();

        System.out.print("Vehicle Number: ");

        String carNumber = sc.nextLine();

        System.out.print("Fuel Type: ");

        String fuelType = sc.nextLine();

        Car car = new Car(carBrand, carSpeed, carColour, carNumber, fuelType);

        System.out.println("\nEnter Electric Car Details:");

        System.out.print("Brand: ");

        String ecBrand = sc.nextLine();

        System.out.print("Max Speed: ");

        double ecSpeed = sc.nextDouble();

        sc.nextLine();

        System.out.print("Colour: ");

        String ecColour = sc.nextLine();

        System.out.print("Vehicle Number: ");

        String ecNumber = sc.nextLine();

        System.out.print("Fuel Type: ");

        String ecFuel = sc.nextLine();

        System.out.print("Battery Capacity (kWh): ");

        int batteryCap = sc.nextInt();

        System.out.print("Charge Remaining (%): ");

        int chargeRemain = sc.nextInt();

        ElectricCar eCar = new ElectricCar(ecBrand, ecSpeed, ecColour, ecNumber, ecFuel, batteryCap, chargeRemain);

        System.out.println("\n==================== OUTPUT ====================");

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

        mv.displayInfo();

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

        car.displayInfo();

        car.displayInfo(true);

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

        eCar.displayInfo();

        eCar.displayInfo(true, chargeRemain);

        sc.close();

    }

}

A white paper with writing on it

AI-generated content may be incorrect.

**OUTPUT:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A black and white page with text

AI-generated content may be incorrect.**

A paper with writing on it

AI-generated content may be incorrect.A paper with writing on it

AI-generated content may be incorrect.

**CODE:**

import java.util.Scanner;

interface Payable {

    void pay();

    String getPaymentDetails();

    boolean validatePayment();

    void generateReceipt();

}

abstract class Payment implements Payable {

    protected double amount;

    public Payment(double amount) {

        this.amount = amount;

    }

    public boolean validatePayment() {

        return amount > 0;

    }

}

class CashPayment extends Payment {

    public CashPayment(double amount) {

        super(amount);

    }

    public void pay() {

        System.out.println("Cash Payment of ₹" + amount + " received.");

    }

    public String getPaymentDetails() {

        return "Payment Type: Cash, Amount: ₹" + amount;

    }

    public void generateReceipt() {

        System.out.println("\n\n=== Receipt ===");

        System.out.println("Payment Method: " + this.getClass().getSimpleName());

        System.out.println("Amount Paid: ₹" + amount);

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

    }

}

class CreditCardPayment extends Payment {

    private String cardNumber;

    private String cardHolder;

    public CreditCardPayment(double amount, String cardNumber, String cardHolder) {

        super(amount);

        this.cardNumber = cardNumber;

        this.cardHolder = cardHolder;

    }

    public void pay() {

        System.out.println("Credit Card Payment of ₹" + amount + " received from " + cardHolder);

    }

    public String getPaymentDetails() {

        return "Payment Type: Credit Card, Card Holder: " + cardHolder + ", Card No: " + cardNumber;

    }

    public void generateReceipt() {

        System.out.println("=== Receipt ===");

        System.out.println("Payment Method: " + this.getClass().getSimpleName());

        System.out.println("Amount Paid: ₹" + amount);

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

    }

}

class UPIPayment extends Payment {

    private String upiID;

    public UPIPayment(double amount, String upiID) {

        super(amount);

        this.upiID = upiID;

    }

    public void pay() {

        System.out.println("UPI Payment of ₹" + amount + " sent to " + upiID);

    }

    public String getPaymentDetails() {

        return "Payment Type: UPI, UPI ID: " + upiID + ", Amount: ₹" + amount;

    }

    public void generateReceipt() {

        System.out.println("\n\n=== Receipt ===");

        System.out.println("Payment Method: " + this.getClass().getSimpleName());

        System.out.println("Amount Paid: ₹" + amount);

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

    }

}

class OverloadDemo {

    public void printDetails(String details) {

        System.out.println(details);

    }

    public void printDetails(String details, double amount) {

        System.out.println(details + " | Amount: ₹" + amount);

    }

    public void printDetails(Payment payment) {

        System.out.println("Details: " + payment.getPaymentDetails());

    }

}

public class PaymentSystem {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        CashPayment cash = new CashPayment(500);

        CreditCardPayment card = new CreditCardPayment(1200, "1234567890123456", "Alice");

        UPIPayment upi = new UPIPayment(750, "alice@upi");

        Payment[] payments = {cash, card, upi};

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

        for (Payment p : payments) {

            if (p.validatePayment()) {

                p.pay();

                p.generateReceipt();

                System.out.println(p.getPaymentDetails());

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

            } else {

                System.out.println("Invalid Payment");

            }

        }

        System.out.println("\n--- Method Overloading Demo ---");

        OverloadDemo od = new OverloadDemo();

        od.printDetails("Simple message");

        od.printDetails("Payment Done", 1500);

        od.printDetails(card);

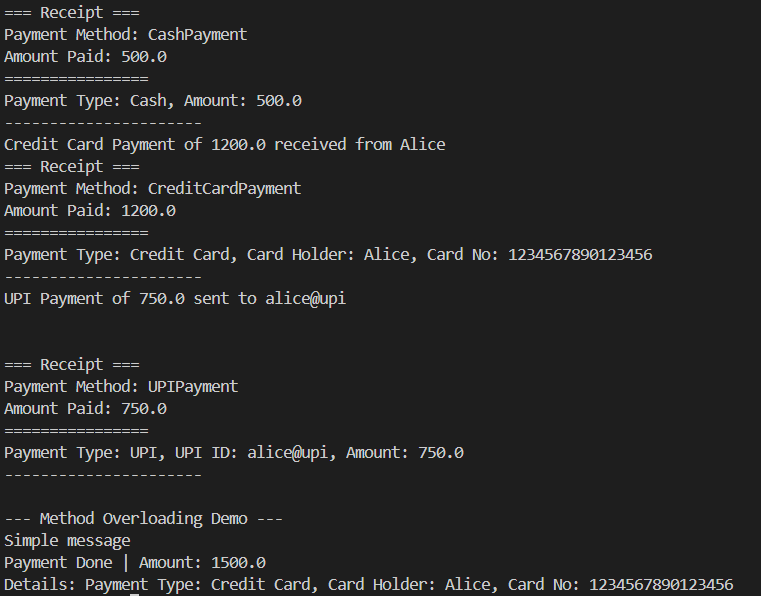
    }

}

A paper with writing on it

AI-generated content may be incorrect.

OUTPUT:



LEARNING OUTCOMES:

* LEARNED TO IMPLEMENT INTERFACE
* LEARNED TO USE ABSTRACT CLASS
* LEARNED TO USE SUPER() IN RIGHT PLACE