Name: Somnath R. Shintre Roll No:

Class: TE CSE Batch:

**Title: -** Create Vehicle Interface with name, maxPassanger, and maxSpeed variables. Create LandVehicle and SeaVehicle Inteface from Vehicle interface. LandVehicle has numWheels variable and drive method. SeaVehicle has displacement variable and launch method. Create Car class from LandVehicle, HoverCraft from LandVehicle and SeaVehicle interface. Also create Ship from SeaVehicle. Provide additional methods in HoverCraft as enterLand and enterSea. Similarly provide other methods for class Car and Ship. Demonstrate all classes in a application.

**Program:-**

/\*\*

 \* MAIN vehicle interface

 \*/

interface Vehicle {

    // Instance variable

public static final String name = "Nissan GTR";

    public static final int maxPassanger = 7;

    public static final int maxSpeed = 160;

}

/\*\*

 \* LandVehicle interface extends vehicle interface

 \*/

interface LandVehicle extends Vehicle {

    // Instance Vatiable

    public static final int numWheels = 4;

    // Drive method

    public abstract void drive();

}

/\*\*

 \* SeaVehicle interface extends vehicle interface

 \*/

interface SeaVehicle extends Vehicle {

    // Instance Vatiable

    public static final int displacement = 500;

    // Drive method

    public abstract void launch();

}

/\*\*

 \* Car class implements LandVehicle interface

 \*/

class Car implements LandVehicle {

    // Overiding the abstract drive method from LandVehicle interface

    @Override

    public void drive() {

        System.out.println(name + " is a manual car with maximun speed " + maxSpeed + "km/hr");

    }

    // Additional Methods

    public void door() {

        System.out.println("It's 4 door SUV with total passanger capacity of " + maxPassanger + " Passangers");

    }

    public void color() {

        System.out.println("The colour of car is 'Dark Blue'");

    }

}

/\*\*

 \* HoverCraft class implements LandVehicle interface & SeaVehicle interface

 \*/

class HoverCraft implements LandVehicle, SeaVehicle {

    // Overiding the abstract drive method from LandVehicle interface

    @Override

    public void drive() {

        System.out.println(

                "HoverCraft several engines usually drives the fan (or impeller), which is responsible for lifting the vehicle by forcing high pressure air under the craft.");

    }

    // Overiding the abstract launch method from SeaVehicle interface

    @Override

    public void launch() {

        System.out.println(

                "The launch type of this ship is  Gravitational type launching with displacement of 3.25 tons");

    }

    // Additional Methods

    public void enterLand() {

        System.out.println("Hovercraft is entering on land surface!");

    }

    public void enterSea() {

        System.out.println("Hovercraft is entering on sea surface!");

    }

}

/\*\*

 \* Ship class implements SeaVehicle interface

 \*/

class Ship implements SeaVehicle {

    // Overiding the abstract launch method from SeaVehicle interface

    @Override

    public void launch() {

        System.out.println("The launch type of this ship is  Gravitational type launching with displacement of "

                + displacement + "tons");

    }

    // Additional Methods

    public void lenght() {

        System.out.println("The Length is 150 m (492.126 ft)");

    }

    public void room() {

        System.out.println("The ship content 50 rooms in total");

    }

}

public class Vehicle\_Interface {

    public static void main(String[] args) {

        // Creating objects or instances

        Car C1 = new Car();

        HoverCraft H1 = new HoverCraft();

        Ship S1 = new Ship();

        System.out.println("\nDemonstration of Car class");

        C1.drive();

        C1.door();

        C1.color();

        System.out.println("\nDemonstration of HoverCraft class");

        H1.drive();

        H1.launch();

        H1.enterLand();

        H1.enterSea();

        System.out.println("\nDemonstration of Ship class");

        S1.launch();

        S1.lenght();

        S1.room();

    }

}

**Output:-**

