Week 8 - S8 - Core OOP - Abstract Class And Interface - Assignment Problem(HW)

**Name:**Ramesh Harisabapathi Chettiar

**Date of Submission:**07/10/25

**QNO1🡪**

Create an abstract class Shape with abstract methods area() and perimeter(). Provide a

concrete method displayInfo().

Create subclasses Circle and Rectangle that implement the abstract methods. Test the

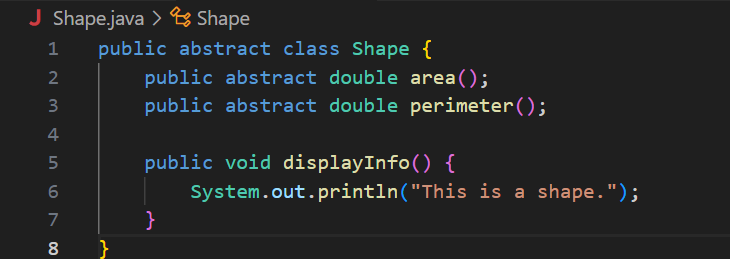
implementation by creating objects and displaying results.

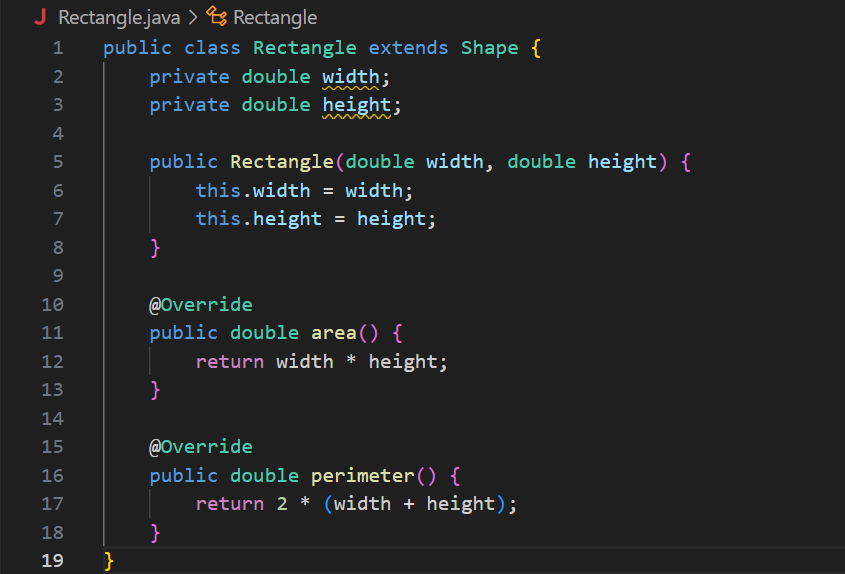
Hints:

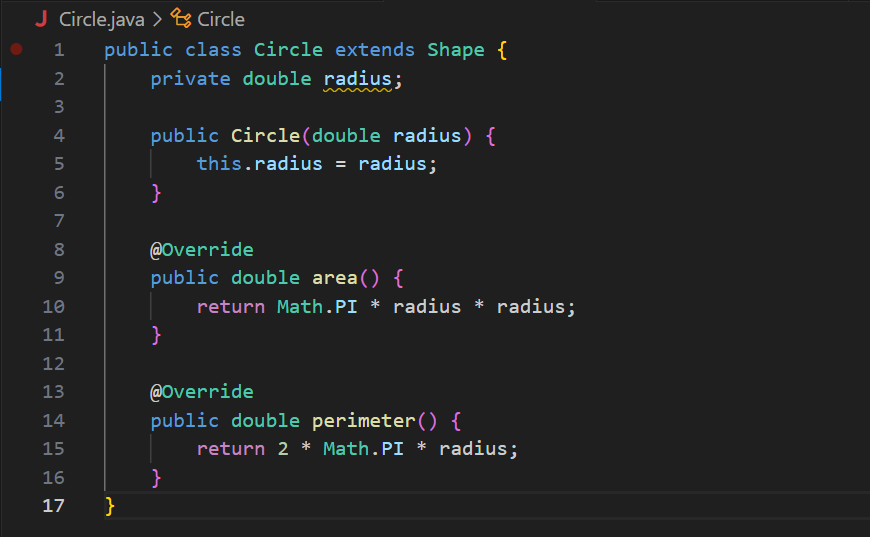
● Use abstract keyword for Shape class.

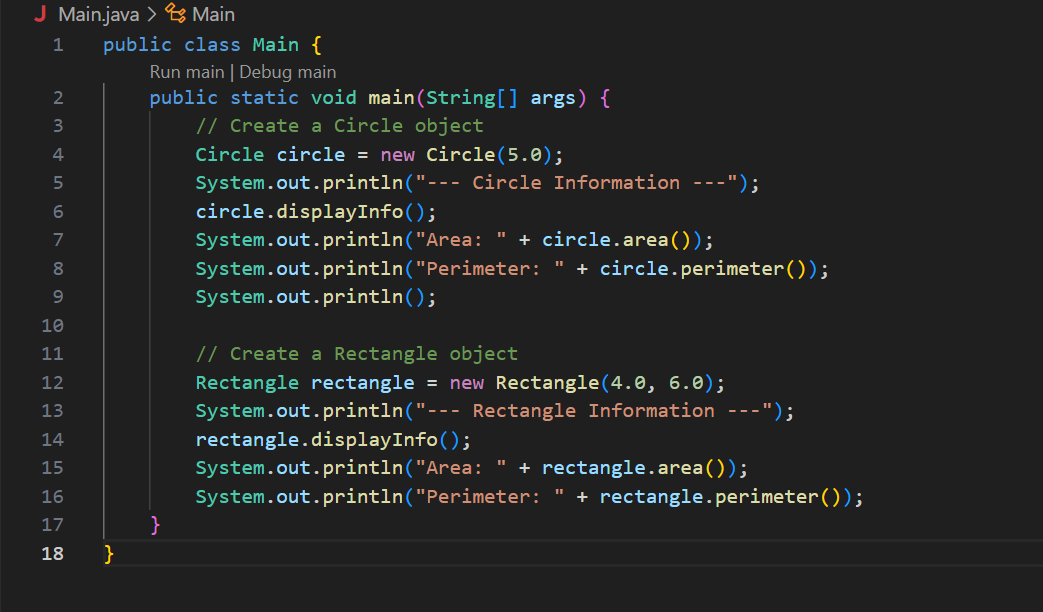
● Implement area() and perimeter() in subclasses.

● Call displayInfo() from subclass objects.

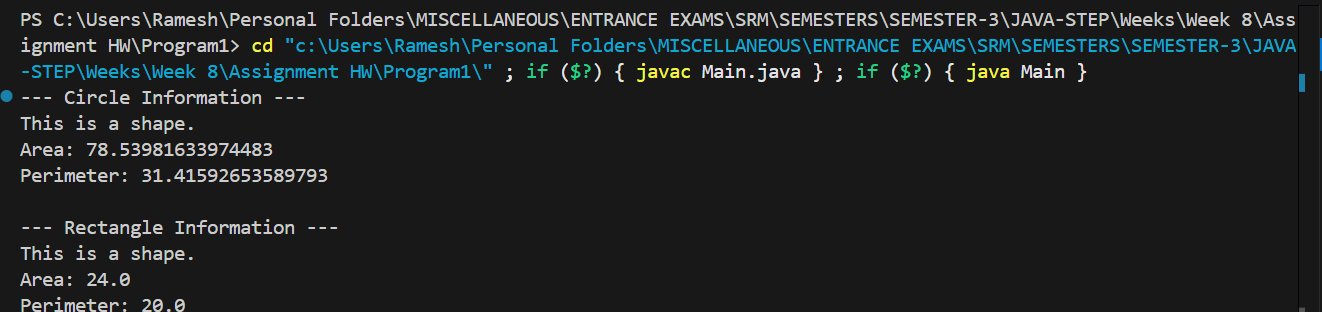
Shape.java

Rectangle.java

Circle.java

Main.java

**OUTPUT🡪**

****

**QNO2🡪**

Create an interface Playable with methods play() and pause().

Create two classes MusicPlayer and VideoPlayer that implement this interface.

Demonstrate polymorphism by storing objects in a Playable reference and invoking methods.

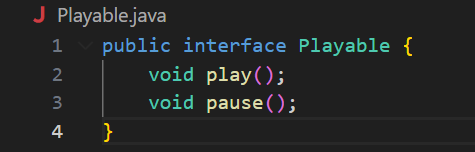
Hints:

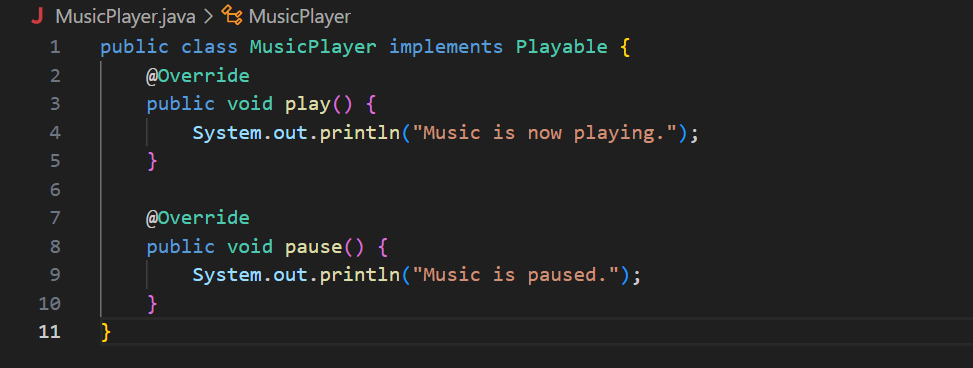
● Use interface keyword.

● Implement both methods in each class.

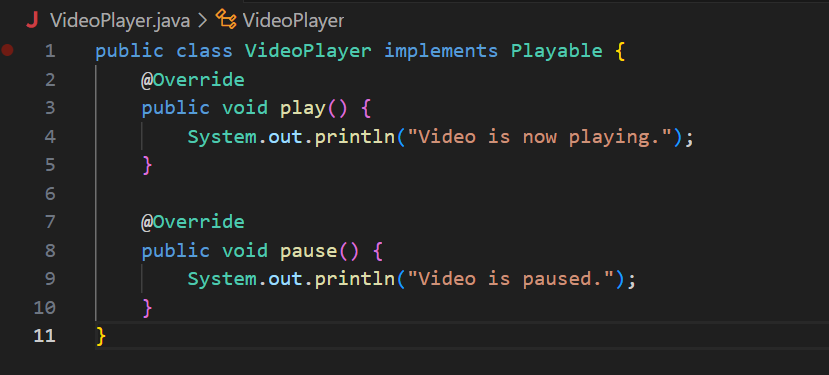
● Use Playable ref = new MusicPlayer(); to test polymorphism.

Playable.java

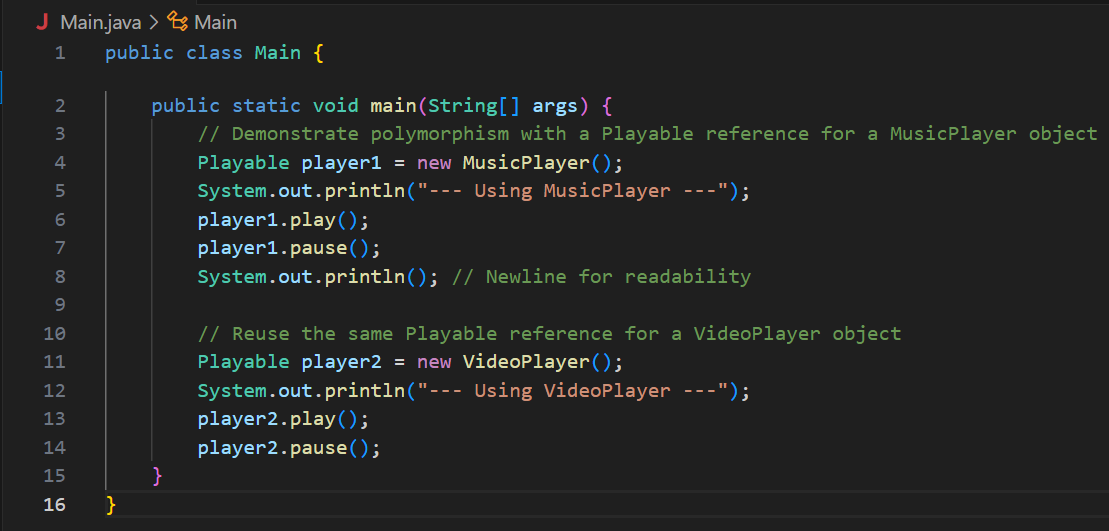
****

MusicPlayer.java

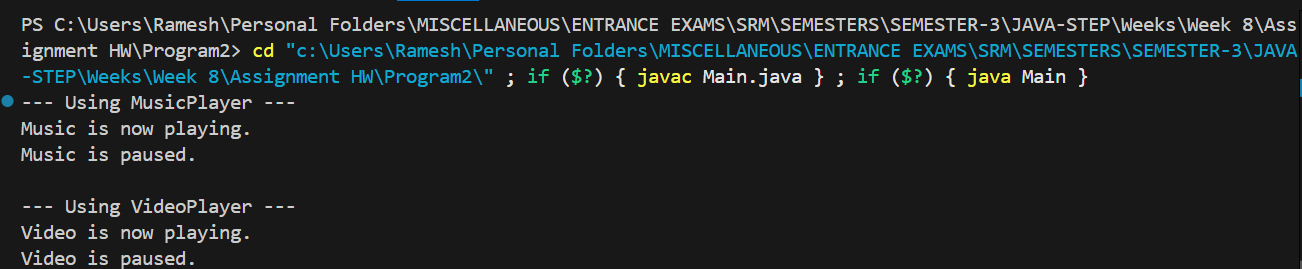
VideoPlayer.java



Main.java



**OUTPUT🡪**

****

**QNO3🡪**

Create an abstract class Vehicle with abstract method start() and a concrete method

stop().

Create an interface Fuel with method refuel().

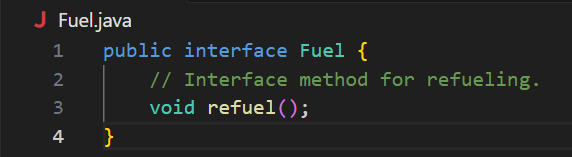
Create class Car that extends Vehicle and implements Fuel. Test all methods.

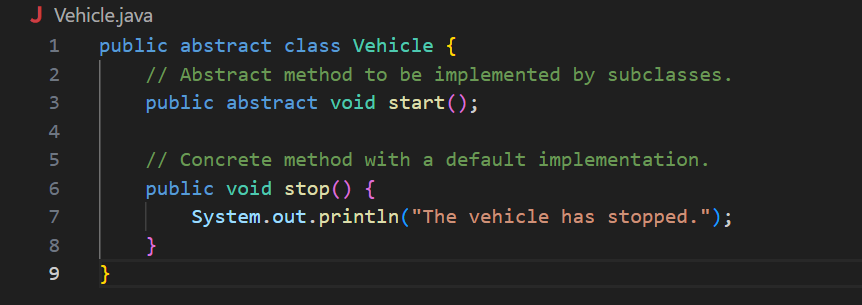
Hints:

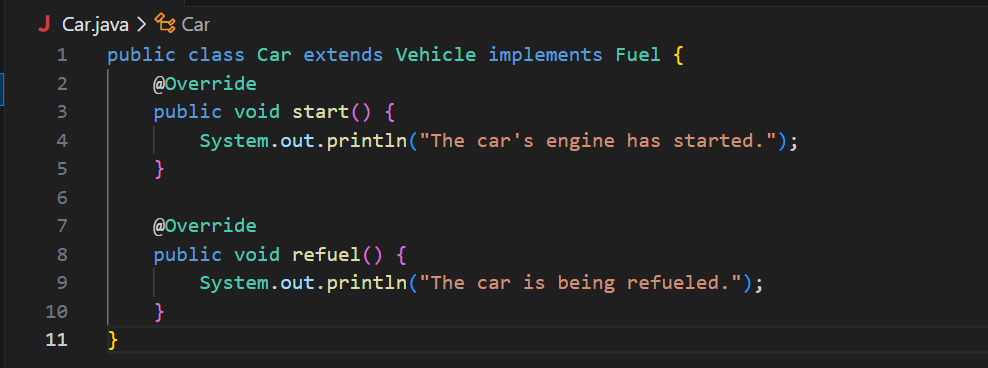
● Use abstract class for Vehicle.

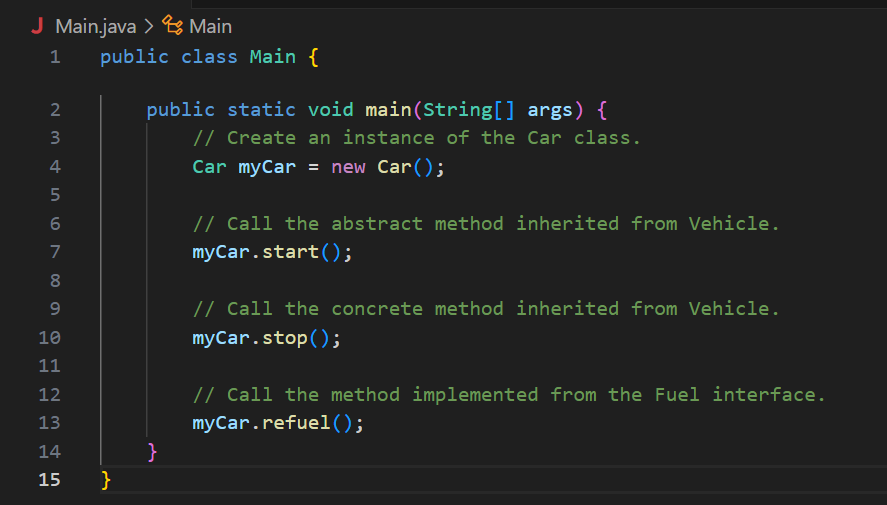
● Implement refuel() from Fuel interface in Car.

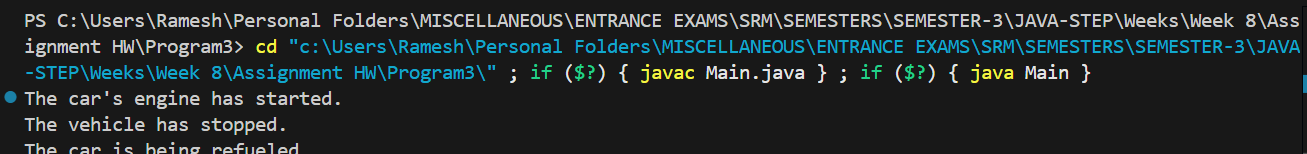
● Show method calls of start(), stop(), and refuel().

Fuel.java

Vehicle.java

Car.java

Main.java

**OUTPUT🡪**

**QNO4🡪**

Create an interface Animal with method eat().

Create another interface Pet that extends Animal and adds method play().

Create a class Dog that implements Pet. Demonstrate interface inheritance in action.

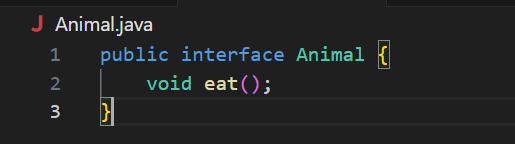
Hints:

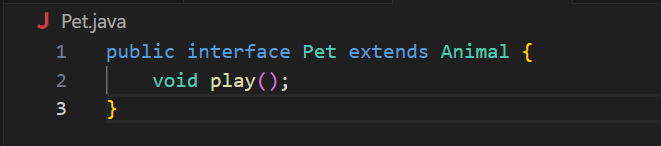
● Use interface Pet extends Animal.

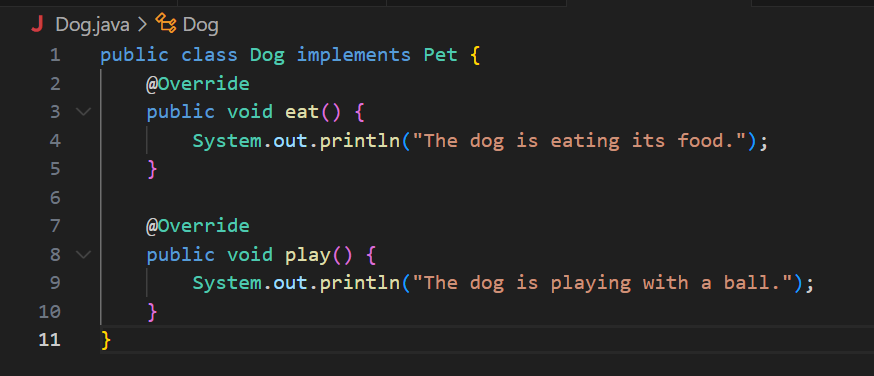
● Dog must implement both eat() and play().

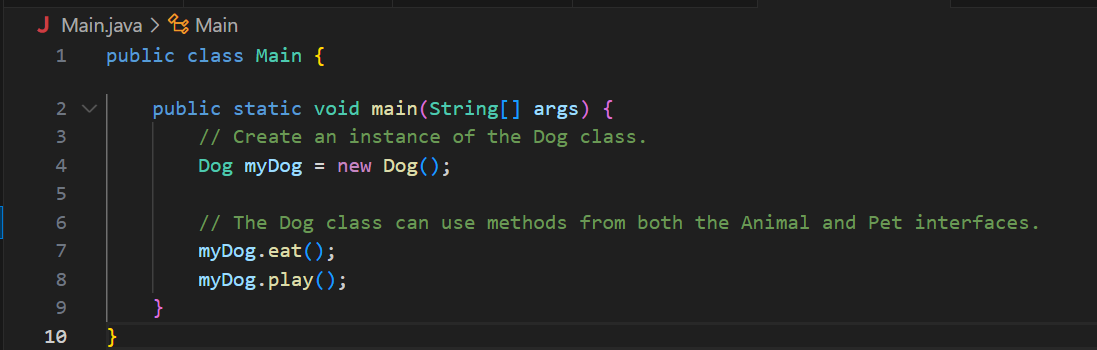
● Create object of Dog and test.

Animal.java

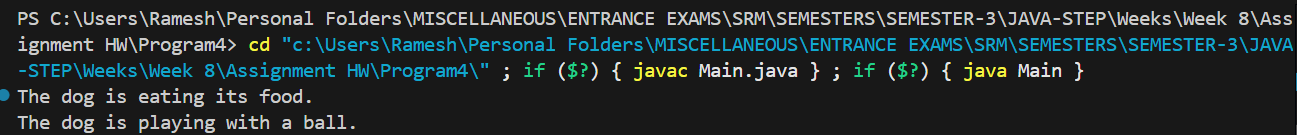


Pet.java

Dog.java

Main.java

**OUTPUT🡪**

****