Problem Set

1. Create an interface Animal with a method sound(). Create two classes Dog and Cat that implement the Animal interface and provide specific implementations for the sound() method. In the main() method, create objects of Dog and Cat and call their sound() methods.
2. Define an interface Shape with a method draw(). Create two classes Circle and Rectangle that implement Shape and provide their own versions of the draw() method. In the main() method, create objects of both Circle and Rectangle and call draw().
3. Create an interface MathConstants with a constant PI (3.14). Create a class Circle that implements MathConstants and calculates the area of a circle using the PI constant. Demonstrate its usage in the main() method.
4. Define an interface Printer with a default method connect() that prints "Printer connected". Create a class LaserPrinter that implements Printer and provides an implementation for print(). Demonstrate the use of the default method in the main() method.
5. Create an interface Vehicle with an abstract method start() and a default method service() that prints "Service required". Implement this interface in the Car class, where start() is implemented. In the main() method, call both start() and service().
6. Create two interfaces Playable and Recordable, each with a method play() and record(). Create a class MediaPlayer that implements both interfaces and provides specific implementations for both methods. Demonstrate how one class can implement multiple interfaces in Java.
7. Create an interface Animal with a method eat(). Then, create another interface Pet that extends Animal and adds a method play(). Create a class Dog that implements Pet and provides implementations for both eat() and play(). Demonstrate interface inheritance in the main() method.
8. Create an interface Calculator with a static method add(int a, int b) that returns the sum of two integers. In the main() method, call the static method directly from the interface without creating an object.
9. Define an interface Payment with a method processPayment(). Create two classes CreditCardPayment and PayPalPayment that implement Payment and provide specific implementations of processPayment(). Demonstrate polymorphism by creating a Payment reference that can point to both CreditCardPayment and PayPalPayment objects.
10. Create two interfaces Printer and Scanner, both with a method connect(). Create a class AllInOnePrinter that implements both Printer and Scanner and resolves the method name conflict by providing a single implementation of connect(). Demonstrate the method resolution in the main() method.

11. Create two interfaces InterfaceA and InterfaceB, both with default methods show(). Create a class ClassC that implements both interfaces and overrides the show() method to resolve the conflict. Demonstrate this in the main() method.
12. Create an interface Appliance with a method start(). Create two classes WashingMachine and AirConditioner that implement Appliance. Write an operateAppliance() method that takes an Appliance object and calls its start() method. Use polymorphism to pass either WashingMachine or AirConditioner objects to operateAppliance().