A) Abstract classes and methods:

```
abstract class Shape {
        abstract double calculateArea();
}

class Circle extends Shape {
        double radius;

        public Circle(double radius) {
                this.radius = radius;
        }

        @Override
        double calculateArea() {
                return Math.PI * radius * radius;
        }
}

class Rectangle extends Shape {
        double length;
        double width;

        public Rectangle(double length, double width) {
                this.length = length;
                this.width = width;
        }

        @Override
        double calculateArea() {
                return length * width;
        }
}

public class Main {
        public static void main(String[] args) {
                Circle circle = new Circle(5);
                Rectangle rectangle = new Rectangle(4, 6);

                System.out.println("Area of Circle: " + circle.calculateArea());
                System.out.println("Area of Rectangle: " +
rectangle.calculateArea());
        }
}
```

B) interfaces

```
interface Shape {
  double calculateArea();
}

class Circle implements Shape {
  double radius;

  public Circle(double radius) {
    this.radius = radius;
  }

  @Override
  public double calculateArea() {
    return Math.PI * radius * radius;
  }
}

class Rectangle implements Shape {
  double length;
  double width;

  public Rectangle(double length, double width) {
    this.length = length;
    this.width = width;
  }

  @Override
  public double calculateArea() {
    return length * width;
  }
}

public class Main {
  public static void main(String[] args) {
    Circle circle = new Circle(5);
    Rectangle rectangle = new Rectangle(4, 6);

    System.out.println("Area of Circle: " + circle.calculateArea());
    System.out.println("Area of Rectangle: " + rectangle.calculateArea());
  }
}
```