EXERCISE 1:

The Point2D class is defined as follows:

Point2D	
+Point2D(x: double, y: double)	Constructs a Point2D object with the specified x- and y-coordinates.
+distance(x: double, y: double): double	Returns the distance between this point and the specified point (x, y).
+distance(p: Point2D): double	Returns the distance between this point and the specified point p.
+getX(): double	Returns the x-coordinate from this point.
+getY(): double	Returns the y-coordinate from this point.
+midpoint(p: Point2D): Point2D	Returns the midpoint between this point and point p.
+distance(p1: Point2D, p2: Point2D): double	Returns the distance between p1 and p2.

The class is partially implemented below. Implement distance(Point2D p1, Point2D p2) and midpoint(Point2D p) and write a test program to test those functionalities:

```
class Point2D {
  private double x;
  private double y;

  public Point2D(double x, double y) {
    this.x = x;
    this.y = y;
  }

  public double distance(double x, double y) {
    return distance(new Point2D(x, y));
  }

  public double distance(Point2D p) {
    return distance(this, p);
  }

  public static double distance(Point2D p1, Point2D p2) {
    // Write code to implement it

  }

  public Point2D midpoint(Point2D p) {
    // Write code to implement it

  }

  public double getX() {
    return x;
  }

  public double getY() {
    return y;
  }
}
```

EXERCISE 2 (INTERFACE):

Create an interface named `Shape` which contains two abstract methods: `calculateArea()` and `calculatePerimeter()`. Then, implement this interface in two classes: `Circle` and `Rectangle`. Define the necessary attributes and methods in each class and provide implementations for the abstract methods declared in the `Shape` interface. Finally, write a simple program to demonstrate the usage of these classes.

EXERCISE 3 (INHERITANCE):

Create a base class called `Vehicle` with attributes `make` and `year` (representing the make and manufacturing year of the vehicle). Then, create two derived classes: `Car` and `Motorcycle`. The `Car` class should have an additional attribute `numDoors`, representing the number of doors, while the `Motorcycle` class should have an additional attribute `engineType`, representing the type of engine (e.g., "electric", "gasoline"). Define necessary constructors and methods to demonstrate inheritance.