***UML CLASS DIAGRAM/OOP PRINCIPLES EXPLANATION***

*A screenshot of a computer

AI-generated content may be incorrect.*

In this diagram we have four classes: **Bicycle, TwoWheeled, Vehicle, and Driver.** These match the code from the CS230\_M1\_assignments.zip for Module 1 (**Bicycle.java, TwoWheeled.java, Vehicle.java, and Driver.java).**

**Bicycle class**

This class has four private attributes: gears (int), cost (double), weight (double), and color (String). They’re marked with a minus sign (–) to show they’re private, meaning they can’t be accessed directly from outside the class. To work with these values, we use methods (getters and setters), which are marked with a plus sign (+) because they are public. Bicycle also has multiple constructors (ways to create a Bicycle object), and some extra methods like outputData() to display information.

**Inheritance**

The inheritance arrow from Bicycle to TwoWheeled shows that Bicycle extends TwoWheeled in other words, a Bicycle is a TwoWheeled vehicle. Similarly, TwoWheeled extends Vehicle, which means all the properties of Vehicle and TwoWheeled are also part of Bicycle. This is the **“is-a”** relationship in object-oriented programming.

**Driver class**

The Driver class just has the main method. Its job is to use the other classes by creating objects and testing them. Because of that, we show dependency arrows (dashed arrows) from Driver to Bicycle, TwoWheeled, and Vehicle. This means Driver depends on those classes to do its work, but it doesn’t inherit from them.