Brooke Slampak

CS-230

3/8/2025

Module 1: UML Class Diagrams

A screenshot of a computer

AI-generated content may be incorrect.

The UML diagram depicted above illustrates four classes and their relationships with one another that demonstrate all four of the Object-Oriented Programming principles: Abstraction, encapsulation, polymorphism, and inheritance.

**Abstraction:**

Abstraction helps minimize code duplication by creating reusable functions or classes that encapsulate the complex logic within a class, allowing developers to call them without needing to understand or manage the underlying details. A programmer working with the Driver class to create Bicycle objects does not need to understand the internal workings of the Bicycle class. Instead, they can interact simply with just its publicly available methods.

**Encapsulation:**

The Bicycle class contains four private variables (gear, cost, weight, color), denoted by the minus (-) sign, meaning they are inaccessible to other classes or functions outside of the Bicycle class. This is an example of encapsulation, since access to these private attributes is restricted. However, the class provides public methods (indicated by the + sign) that use getter and setter methods, which enable controlled access to modify and retrieve the private variables from the Bicycle class.

**Polymorphism:**

The concept of polymorphism is illustrated by the Bicycle class having four constructors with different parameters. Depending on the parameters provided when a Bicycle object is instantiated, the appropriate constructor is called, demonstrating the ability of multiple methods to coexist with different parameter sets as defined by polymorphism.

**Inheritance:**

The open arrows connecting the Bicycle class to the TwoWheeled class and the Vehicle class represent inheritance. Even though the TwoWheeled class and Vehicle class do not contain attributes in this diagram example, the Bicycle class would, in principle, inherit attributes from the TwoWheeled and Vehicle classes.