Method Overriding

Scenario-1

Employee and Manager - Overriding Work Method:

Description: Create an Employee class with a method work(). Then, create a subclass Manager that overrides work() to define specific managerial tasks (e.g., "Managing team" instead of generic "Working").

Real-world scenario: An Employee works generally, but Managers work in more specific ways that override the general employee behaviour.

Scenario-2

Appliance and WashingMachine - Overriding TurnOn Method:

Description: Create a class Appliance with a method turnOn(). Then, create a subclass WashingMachine that overrides the turnOn() method to provide a specific implementation (e.g., starting the wash cycle).

Real-world scenario: Appliances turn on in general, but certain appliances like a WashingMachine may have additional functionality when turned on, which overrides the general method.

Method Overloading

Scenario-3:

**Appliance, Refrigerator, and SmartRefrigerator:**

**Description:** Define a class Appliance with attributes like brand, powerRating, and methods like turnOn() and turnOff(). Create a subclass Refrigerator with additional attributes like temperatureControl and methods like setTemperature(). Then, create a subclass SmartRefrigerator with advanced features like connectToWiFi() and remoteControl().

**Real-world scenario:** A SmartRefrigerator is a type of Refrigerator, which in turn is a type of Appliance.