Question 4:

Design patterns are powerful tools for building reusable and maintainable object-oriented designs, but there are cases where using them can actually make a solution worse. In class, we discussed that patterns are not code but reusable design ideas meant to guide structure, especially when solving recurring problems. However, when a problem is simple or doesn't involve variation, forcing a pattern into the solution can lead to unnecessary complexity. A clear example from the lecture is applying the Template Method pattern to a problem that has only one fixed way of doing things. The Template Method is designed to define the skeleton of an algorithm and let subclasses customize certain steps. But if there's only one implementation, this results in an abstract superclass that's never reused, a small subclass that does all the work, and extra method calls that only add indirection. In these situations, a straightforward method is more readable, easier to maintain, and just as effective. Recognizing when not to use a design pattern is part of writing clean, practical object-oriented code.