Best Practices for Assignment <1>: Abstraction in Java

Summary of Learning:

- **Abstraction Principle:** Abstraction is a fundamental concept in Java that allows you to define the structure of a class without providing a complete implementation. It hides the complex details and exposes only the essential features.

- **Types of Abstraction:**

  Abstract Class: An abstract class is a class that cannot be instantiated and may have both abstract (unimplemented) and concrete (implemented) methods. It serves as a blueprint for derived classes to implement specific behaviors.

  Interface: An interface in Java is a contract that defines a set of abstract methods. Classes that implement an interface must provide concrete implementations for all the methods declared in the interface.

Alternative Ways to Complete the Assignment:
Best Practices:

Abstract Class:

- **Keyword abstract:** The class itself should be declared as abstract using the abstract keyword.
- **Inheritance:** Abstract classes are meant to be extended by concrete subclasses. Subclasses should use the 'extends' keyword to inherit from an abstract class.
- **Avoid Multiple Inheritance:** Java allows a class to inherit from only one abstract class to avoid ambiguity.

Interface:

- **Multiple Interface Implementation:** A class can implement multiple interfaces, separated by commas. However, if multiple interfaces contain methods with the same signature, the class must provide an implementation for that method to avoid ambiguity.
- **Method Signatures:** In an interface, you define method signatures without providing any method implementations. These method signatures consist of the method name, return type, and parameter list (if any).
- **Access Control:** Methods declared in an interface must not have an access modifier other than public. This is because all interface methods are implicitly public.