Week 4 – Use Cases for Polymorphism, Encapsulation, Overrriding and Overloading

Marvin Rogers

**Reservation Class**

The Reservation class is declared as abstract, meaning it cannot be directly instantiated. However, its subclasses (Hotel, Cabin, and House) inherit the abstract methods calculatePricePerNight() and calculateTotalPrice().

Each subclass implements these methods with their specific logic for calculating prices based on their unique attributes. This demonstrates runtime polymorphism, where the appropriate implementation of the method is determined at runtime based on the actual object type (Hotel, Cabin, or House).

**Account Class**

+ getAccount(String accountNumber) // Original method

+ getAccount(String email, String phone) // Overloaded method

The abstract class Reservation declares the methods calculatePricePerNight() and calculateTotalPrice().

The subclasses (Hotel, Cabin, and House) inherit these methods from their parent class.