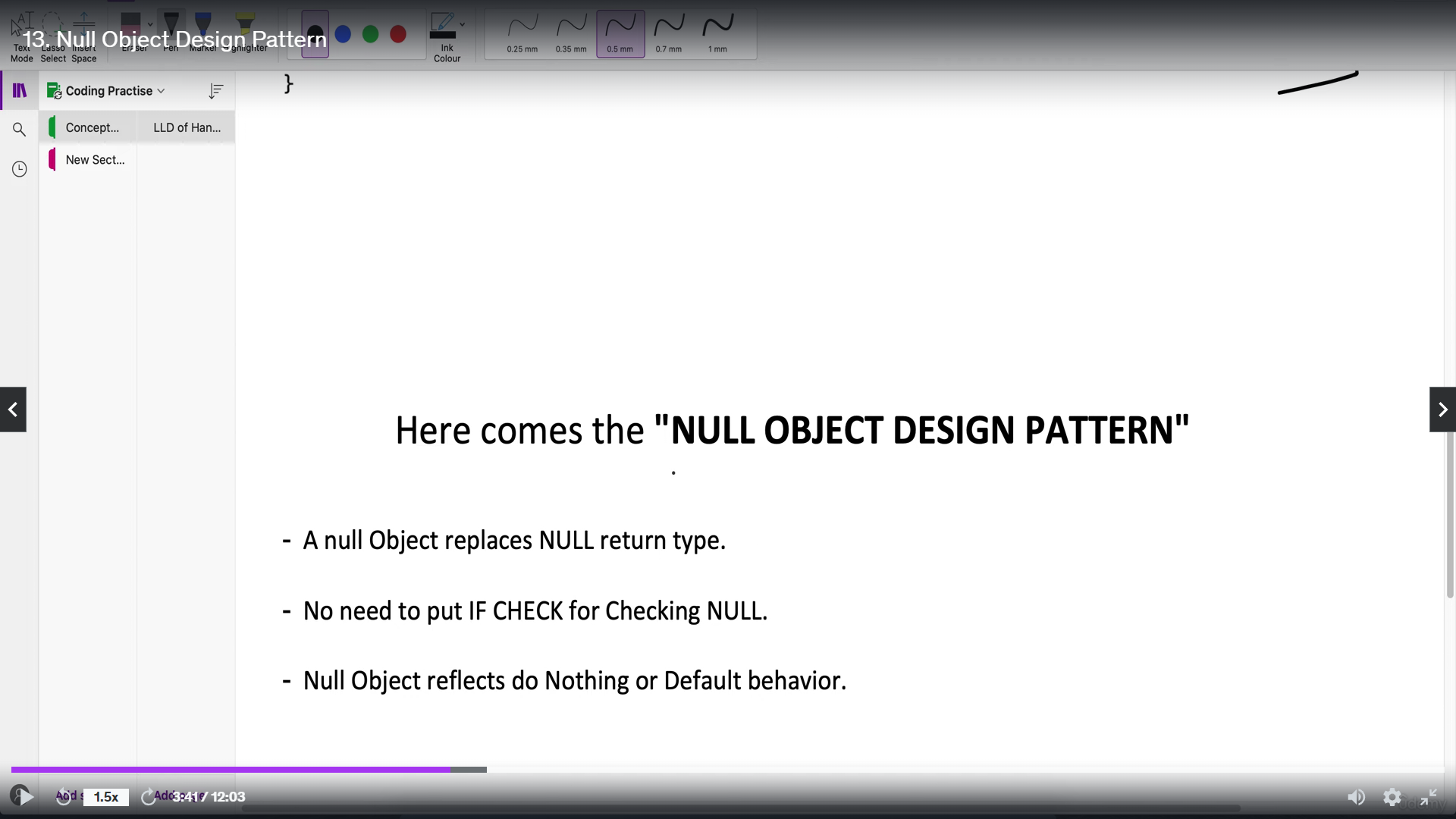
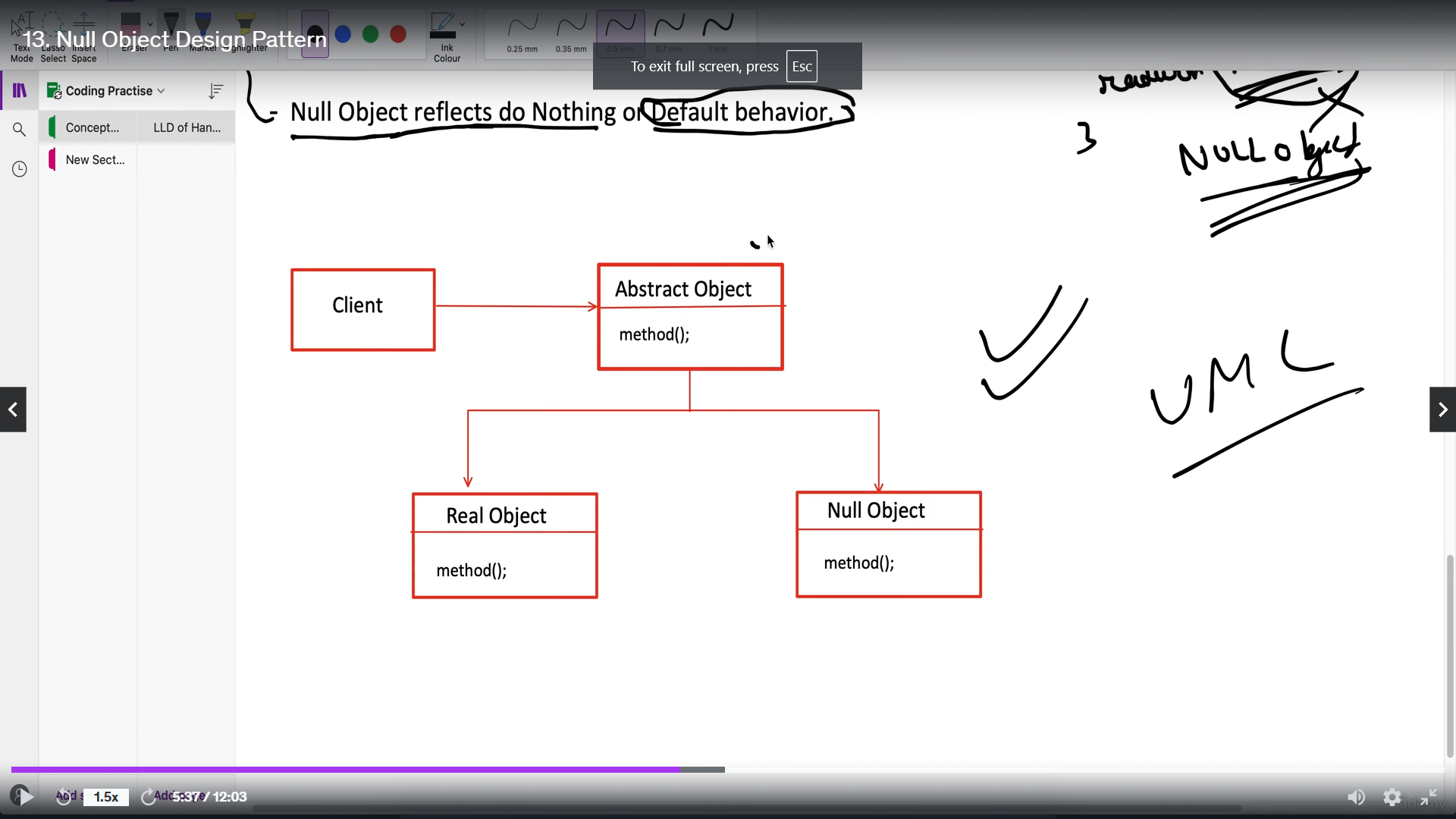
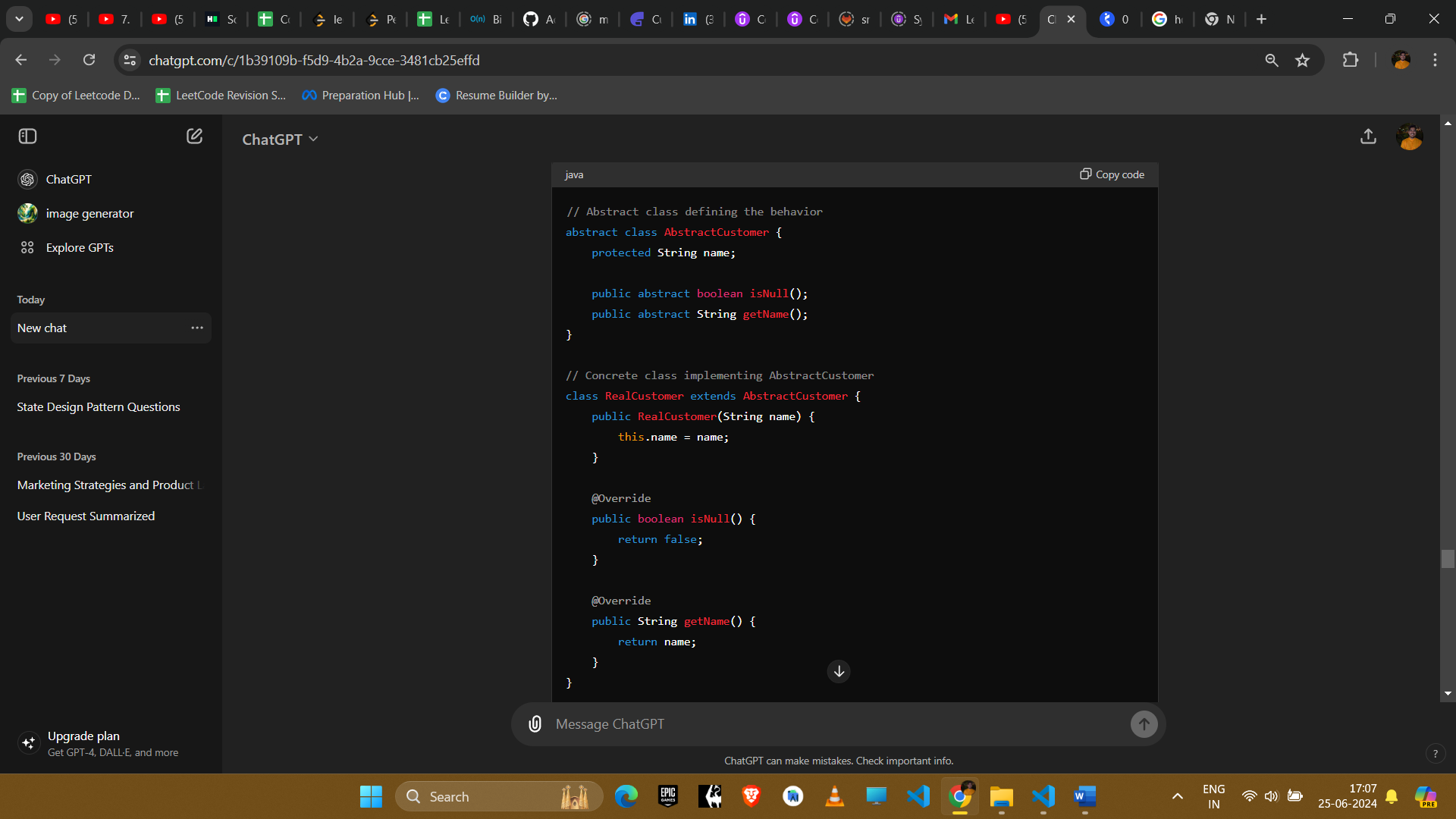
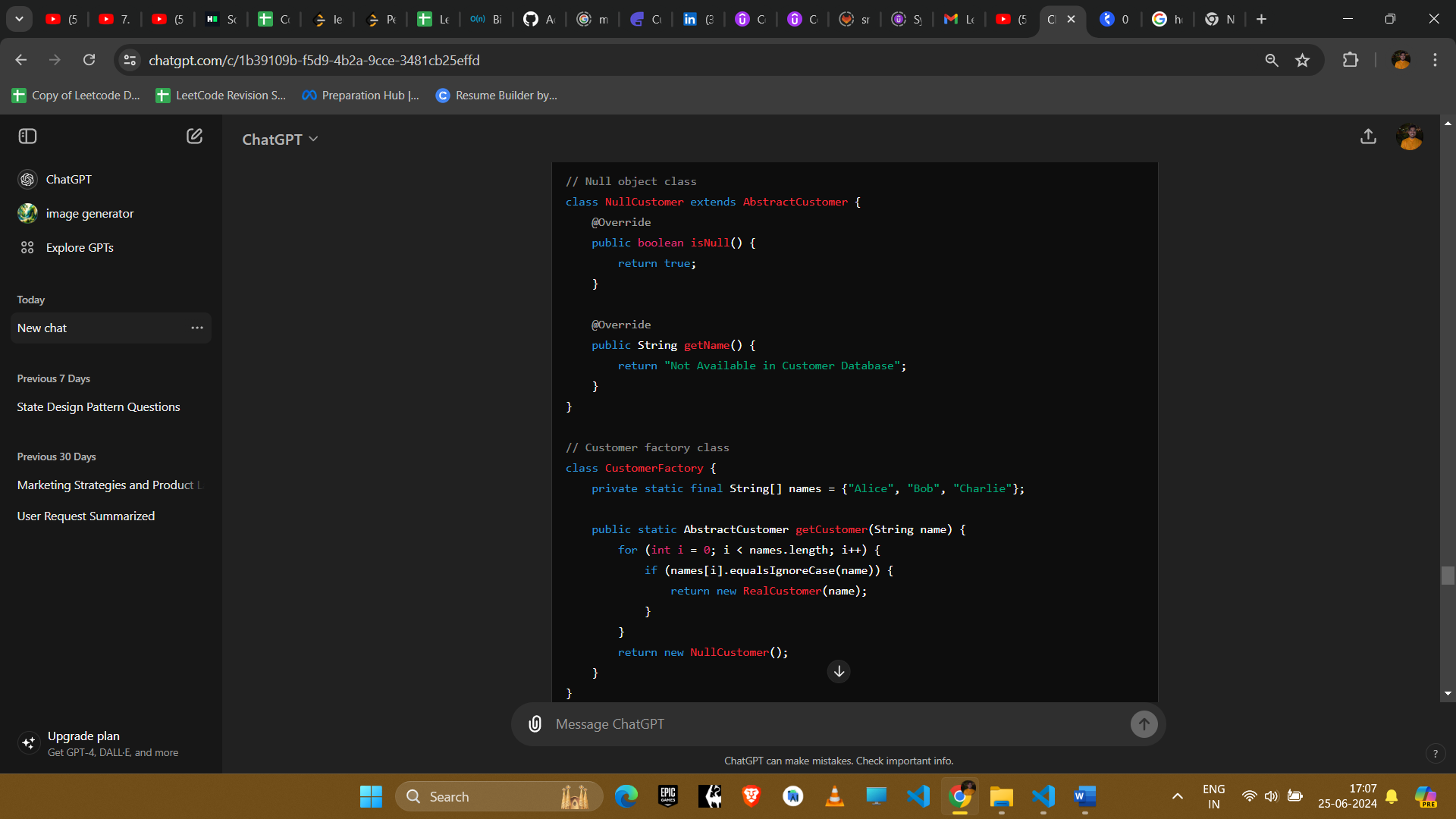
**Null Object Design Pattern**

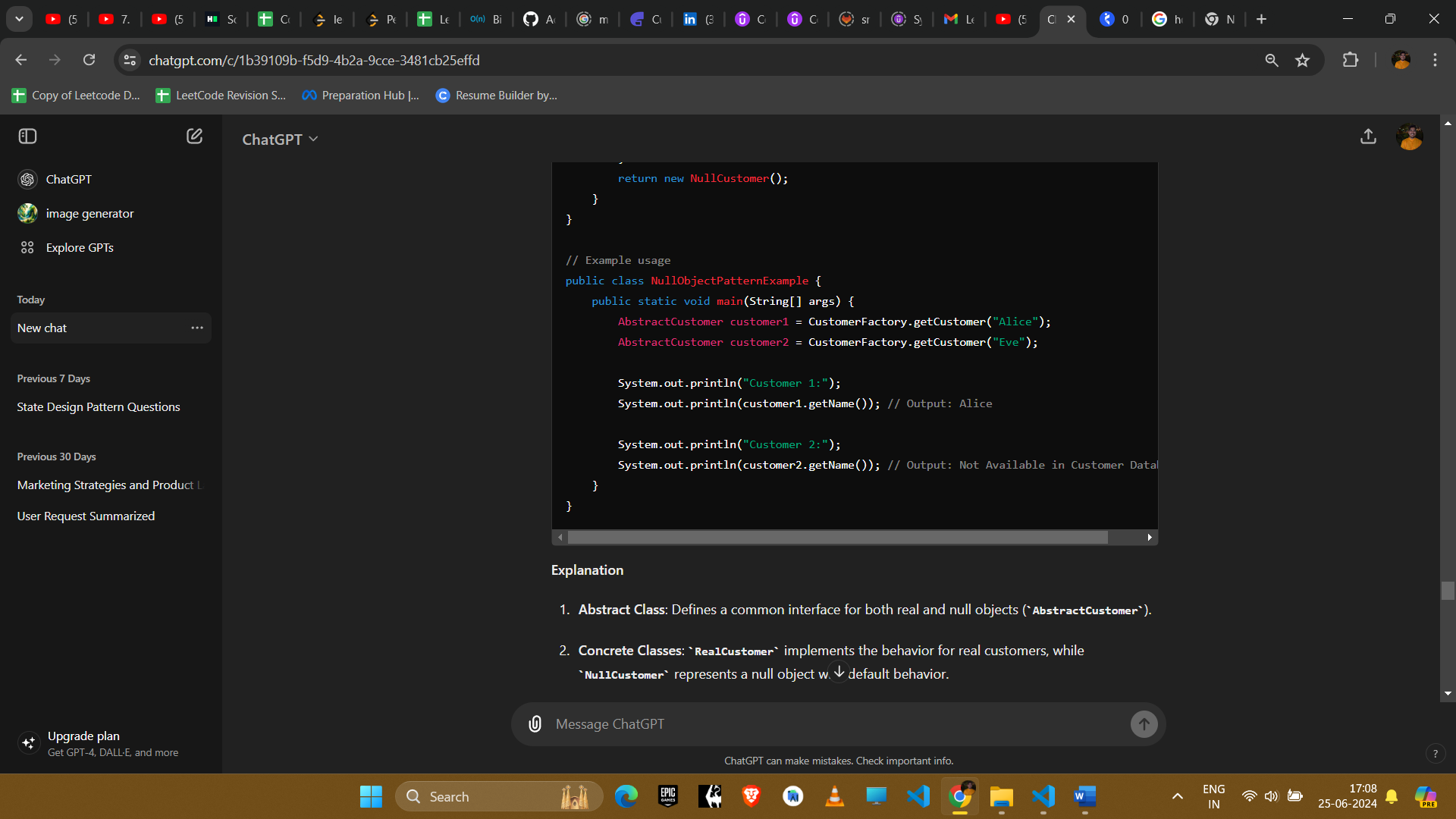
The Null Object design pattern is a behavioral pattern that provides an object with neutral or default behavior to avoid null references in code. The Null Object pattern is a behavioral design pattern where a null object substitutes for a nullable object. Instead of using null references to represent absence of an object, a null object provides a "do nothing" behavior — an object that implements the expected interface, but whose methods have empty bodies or return default values. This approach ensures that the client code can operate without the need for null checks.











#### Explanation

1. **Abstract Class**: Defines a common interface for both real and null objects (AbstractCustomer).
2. **Concrete Classes**: RealCustomer implements the behavior for real customers, while NullCustomer represents a null object with default behavior.
3. **Factory Class**: CustomerFactory creates and returns instances of RealCustomer or NullCustomer based on input.

#### Example Uses in Amazon Interviews

1. **Error Handling**
   * **Scenario**: Handling null references in a safe and predictable manner.
   * **Implementation**: Null objects provide default behavior, preventing null pointer exceptions.
2. **Default Values**
   * **Scenario**: Providing default values when an object is absent or not found.
   * **Implementation**: Null objects return predefined values, ensuring consistent application behavior.
3. **Optional Dependencies**
   * **Scenario**: Handling optional dependencies without conditional checks.
   * **Implementation**: Null objects can be used to represent missing or optional components in a system.

#### Conclusion

The Null Object pattern improves code robustness by replacing null references with objects that provide default behavior. It simplifies logic by eliminating the need for repetitive null checks, enhancing code readability and maintainability. This pattern is particularly useful in scenarios where handling null values or absent objects is critical to maintaining system stability and reliability.