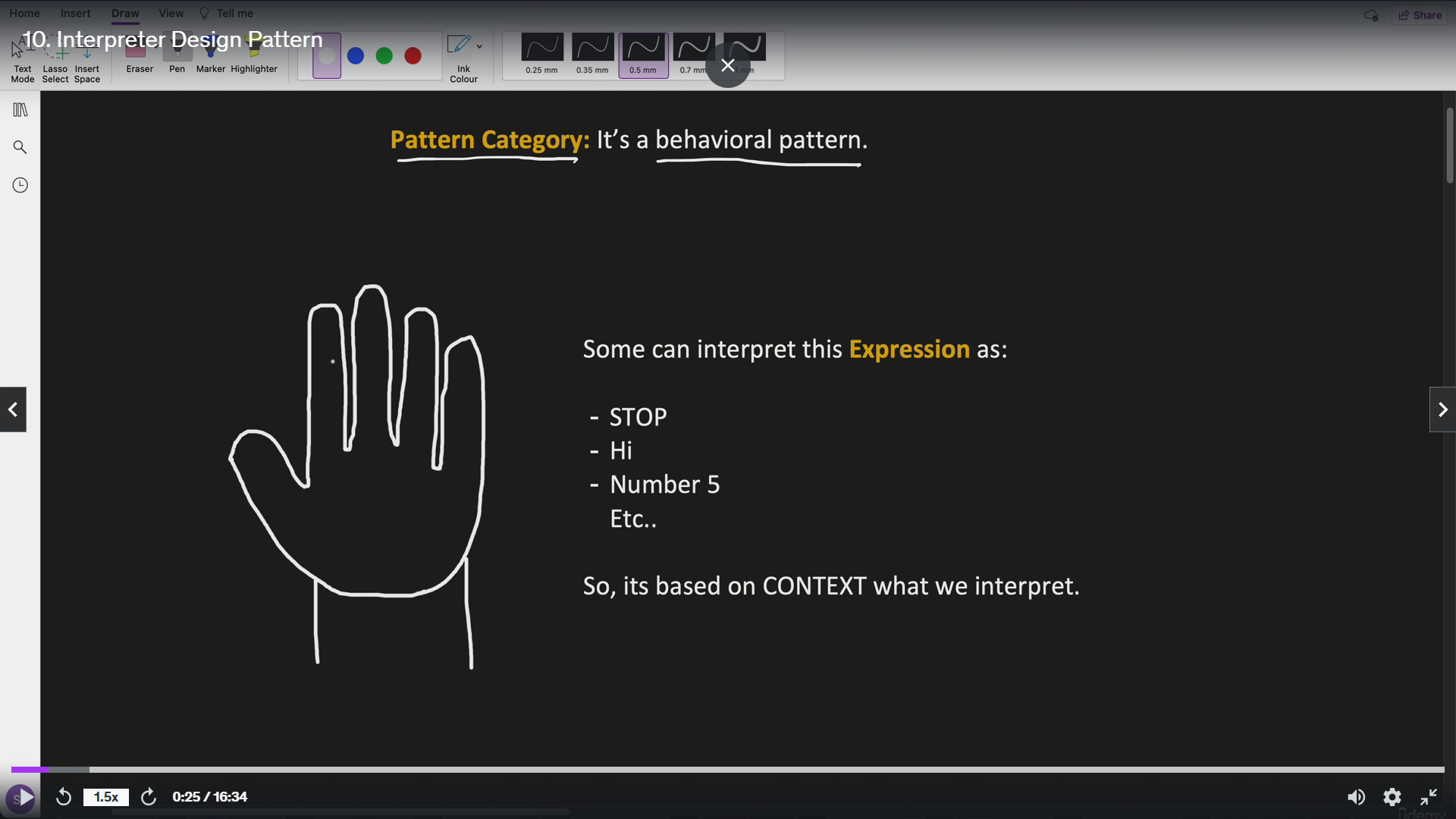
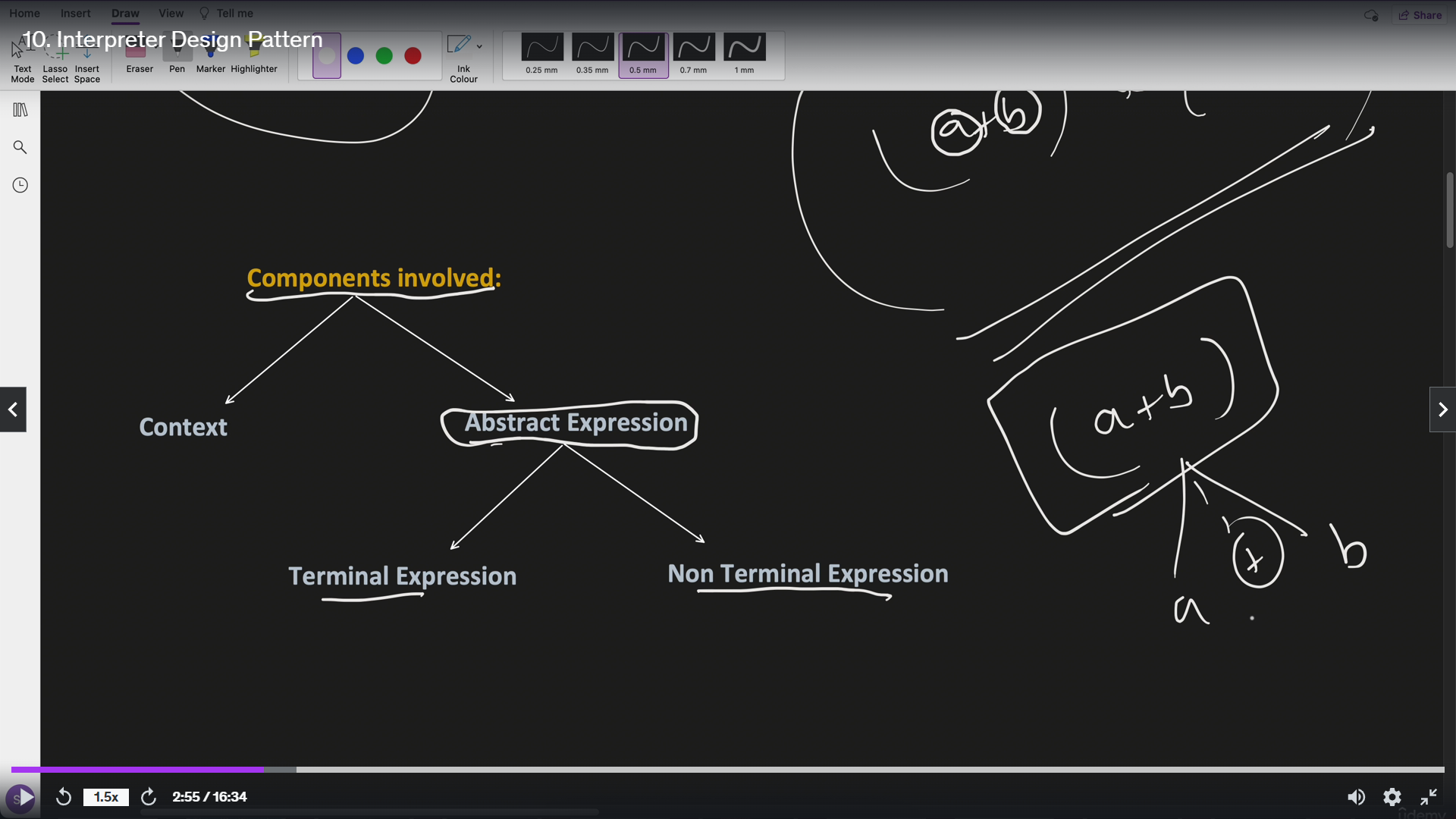
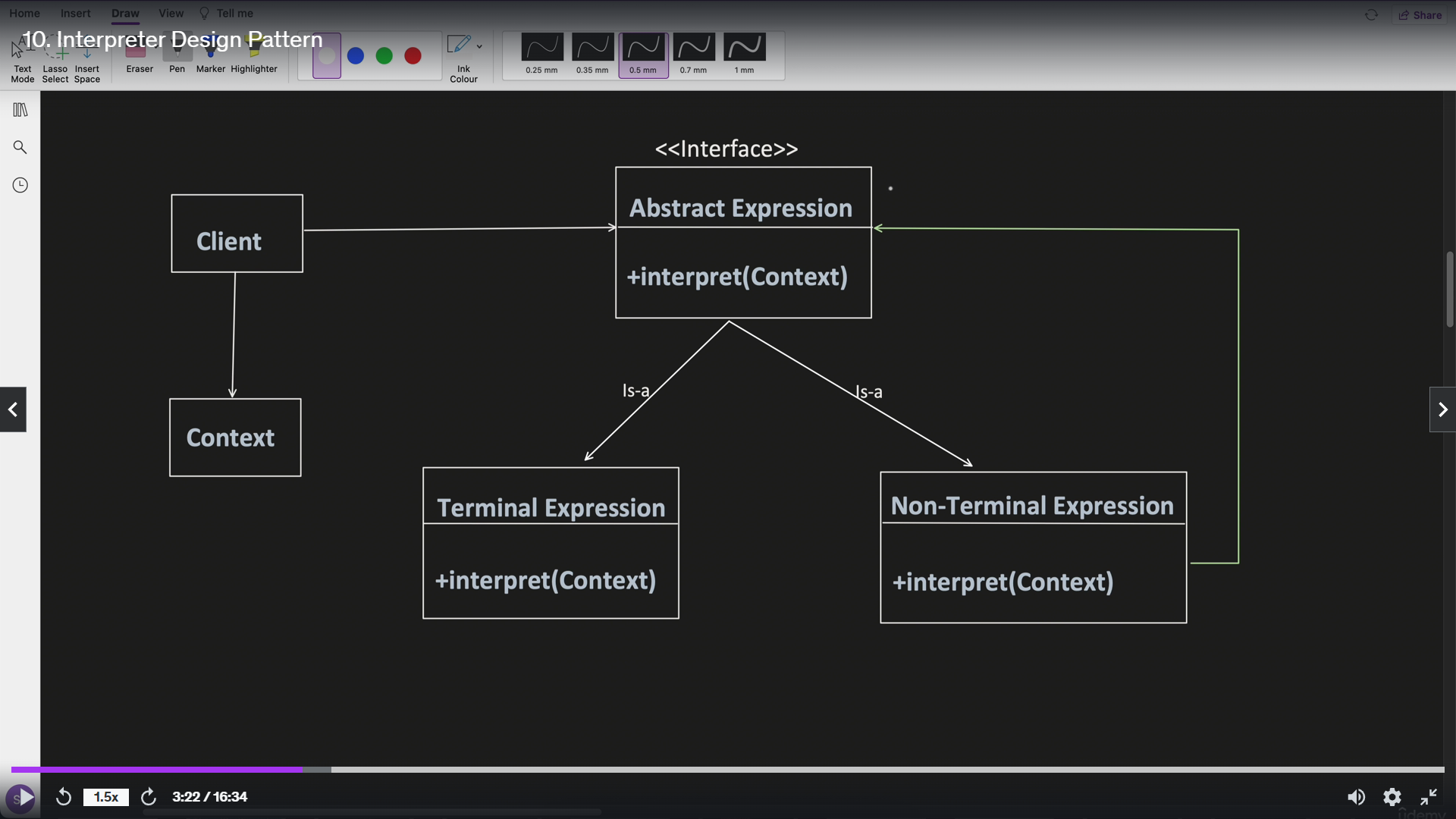
**Interpreter Design Pattern (Solve equations, expressions)**

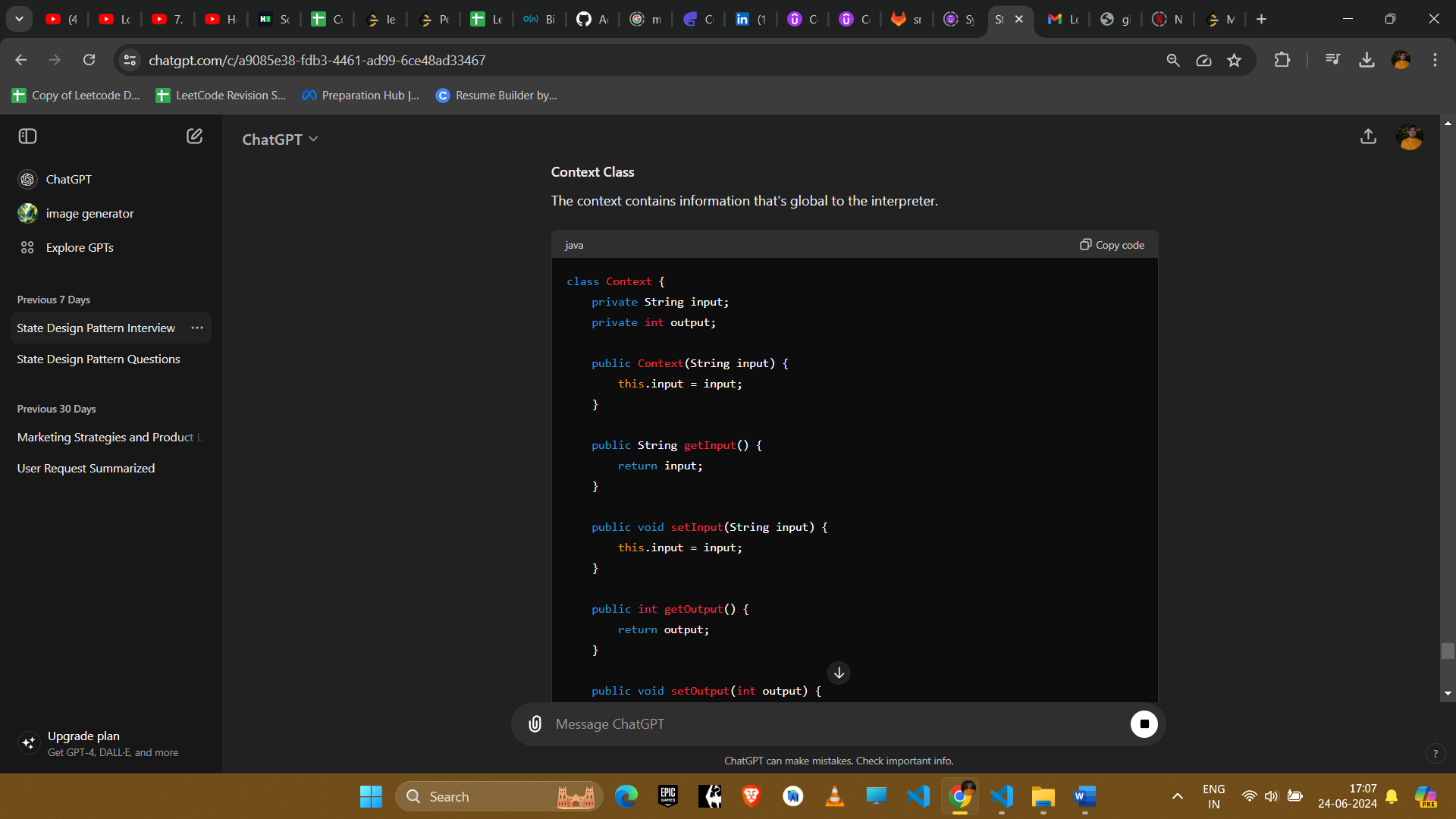
The Interpreter design pattern is a behavioural pattern that defines a representation for a language's grammar along with an interpreter that uses this representation to interpret sentences in the language. This pattern is useful for implementing and processing languages, expressions, or commands.

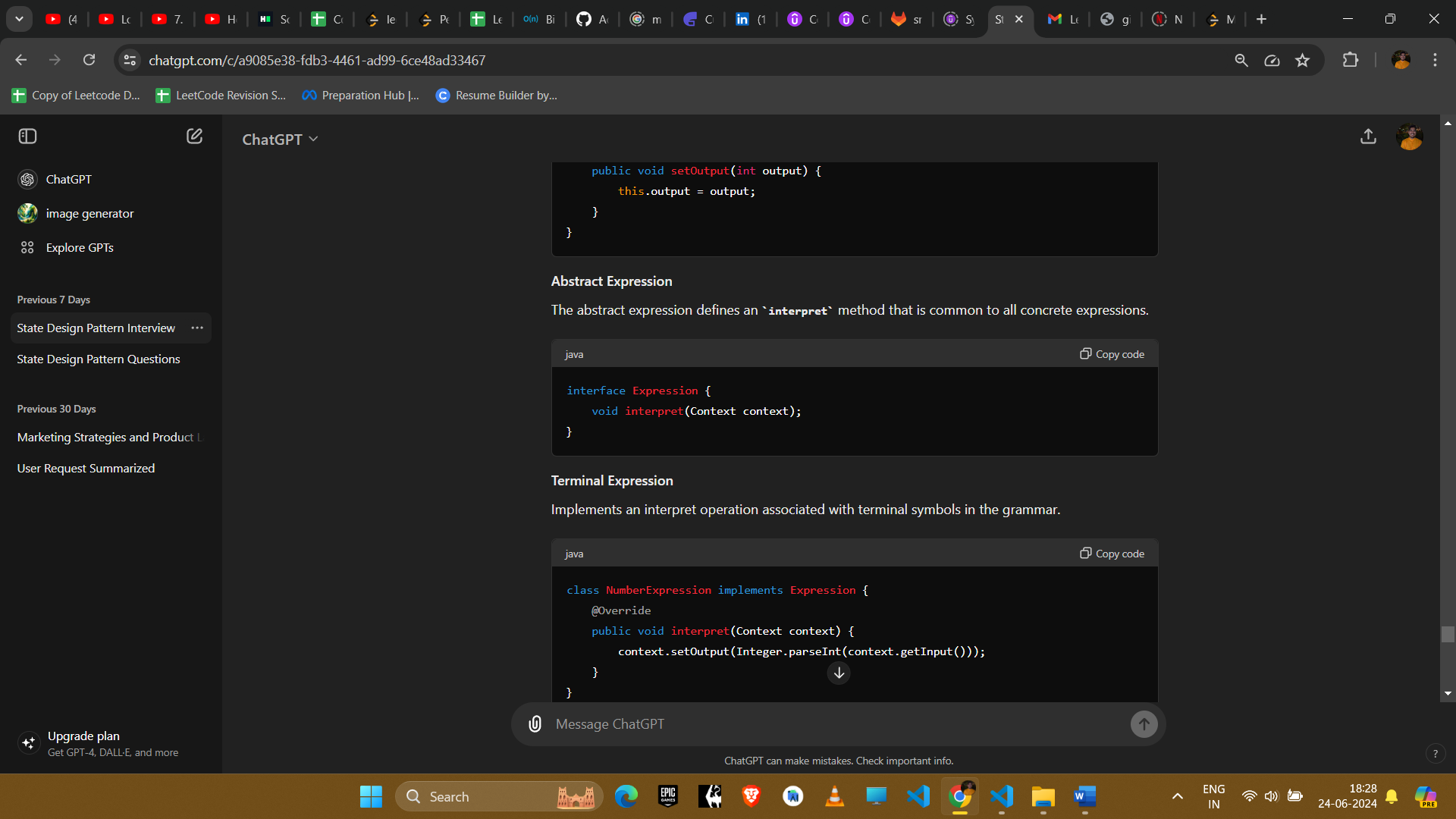
Eg of expressions: multiplication of two numbers, solving an equation, etc.

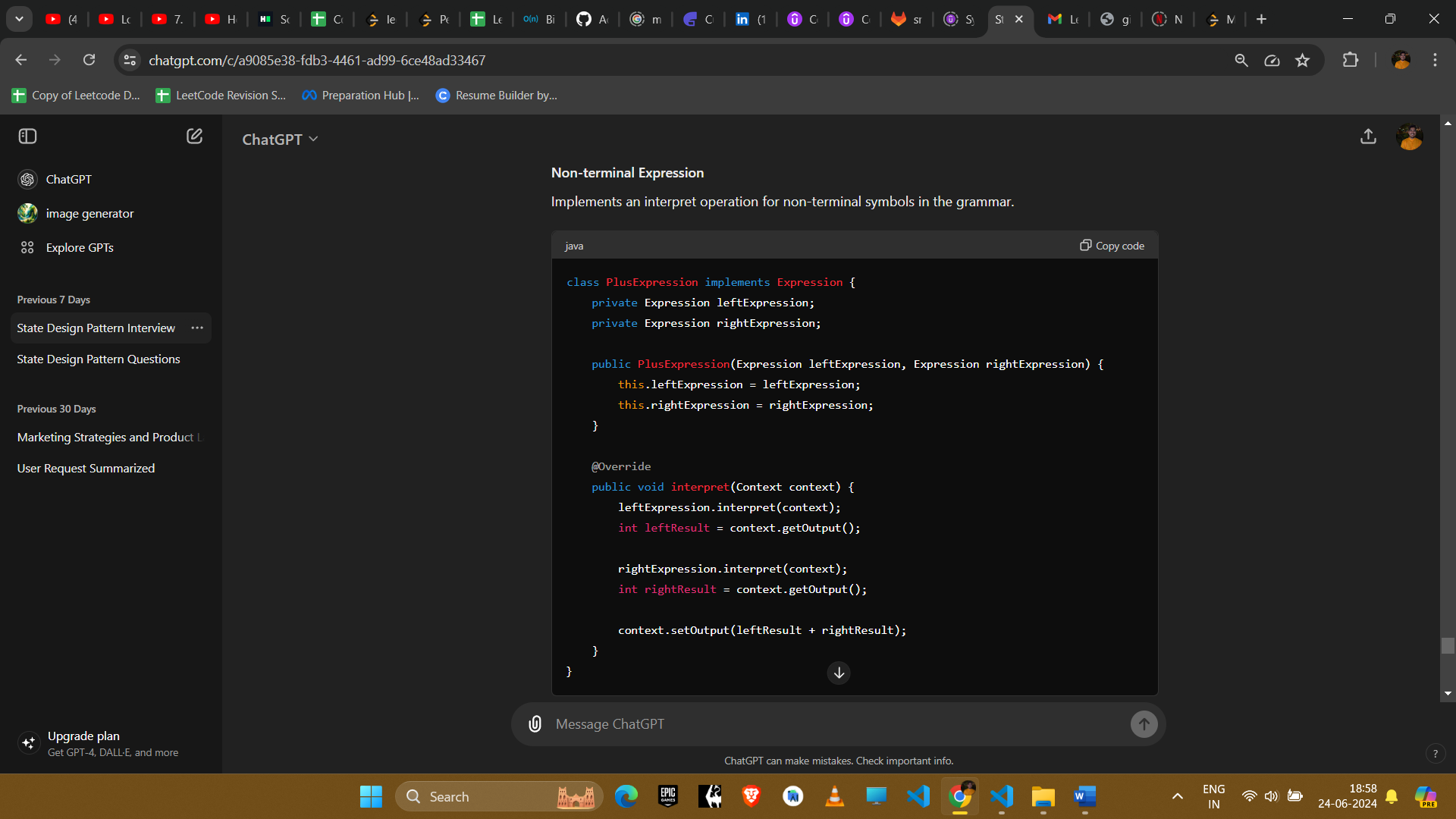


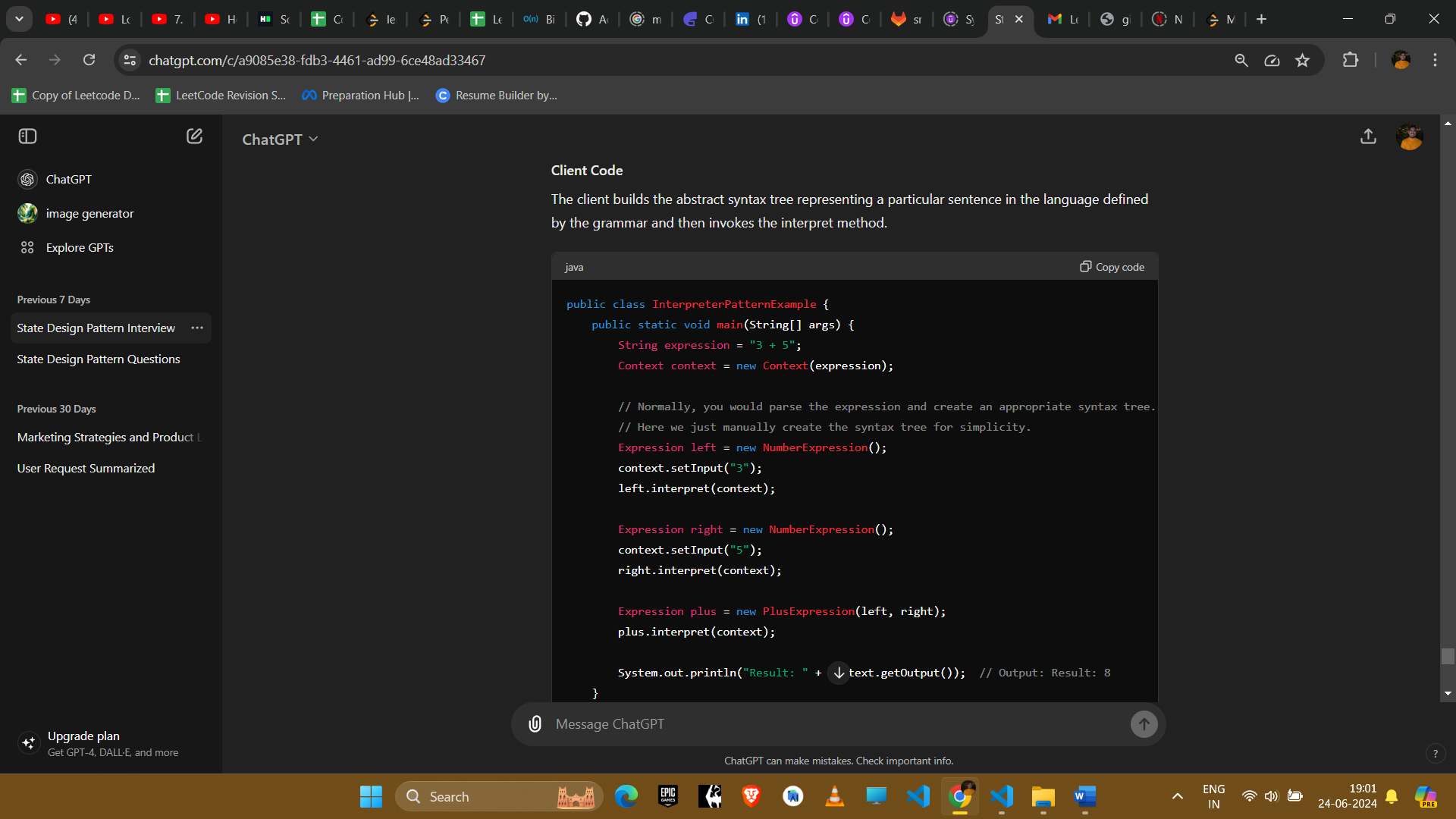












### Explanation

1. **Context**: Contains information that is global to the interpreter. (eg: get or put values in hashmap)
2. **Abstract Expression**: Interface for expressions with an interpret method.
3. **Terminal Expression**: Implements the interpret operation for terminal symbols.
4. **Non-terminal Expression**: Implements the interpret operation for non-terminal symbols (composed of other expressions).

### Example Uses in Amazon Interviews

#### 1. **Configuration File Parsing**

* **Scenario**: Parsing and interpreting configuration files written in a domain-specific language.
* **Implementation**: Use the Interpreter pattern to define the grammar and implement the interpreter for the configuration language.

#### 2. **Expression Evaluation**

* **Scenario**: Evaluating mathematical expressions or logical expressions.
* **Implementation**: Define a grammar for the expressions and use the Interpreter pattern to evaluate them.

#### 3. **Command Processing**

* **Scenario**: Interpreting and executing a sequence of commands in scripting languages.
* **Implementation**: Implement the grammar and interpreter for the command language.

#### 4. **Data Validation**

* **Scenario**: Validating data formats against specified rules.
* **Implementation**: Use the Interpreter pattern to define the rules and interpret data according to these rules.

#### 5. **Language Translation**

* **Scenario**: Translating one language to another or processing natural language commands.
* **Implementation**: Define the grammar and interpreter for the source language to translate or process it.

### Conclusion

The Interpreter design pattern is ideal for scenarios where you need to interpret and process expressions or commands defined by a formal grammar. It is useful for implementing languages, configuration file parsers, expression evaluators, and command processors. This pattern promotes extensibility and makes it easy to define and implement new expressions or commands.