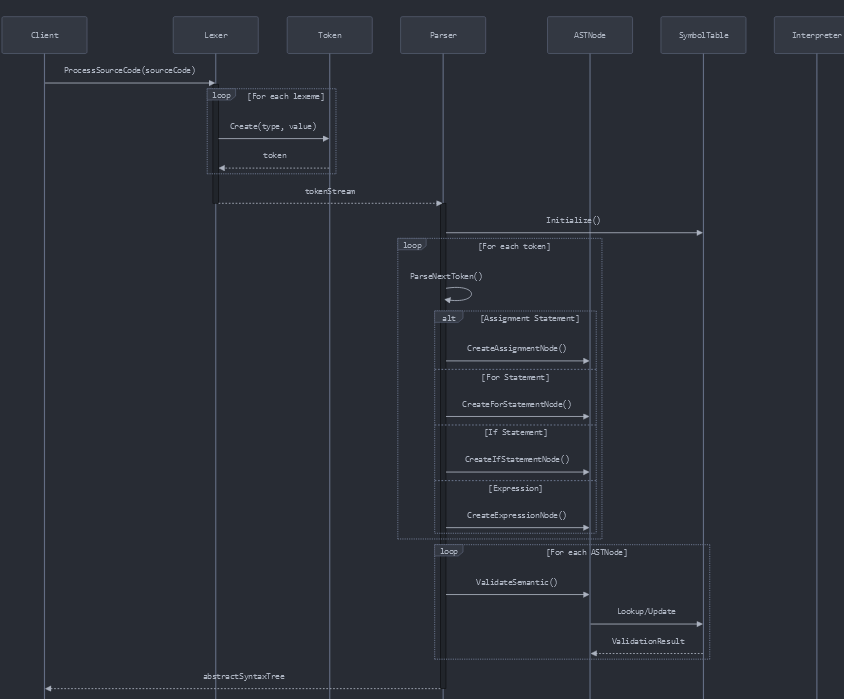
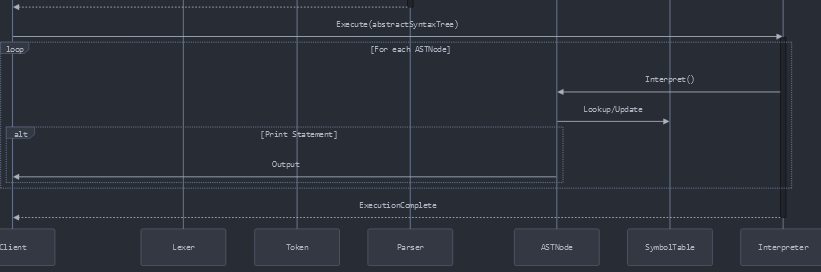
Name:Mohammad Asim Khan  
Reg no:FA21-BCS-037  
Compiler Construction Lab Terminal  
  
Question 1: Class of sequence of mini compiler

Answer:





#### ****1. Lexer Class (****Lexer.cs****)****

* **Purpose:** Tokenizes the input source code.
* **Sequence:**
  + Reads the raw input (source code) provided as a string.
  + Splits the input into tokens (e.g., int, a, =, 3, etc.).
  + Recognizes keywords, identifiers, operators, and literals.
  + Produces a stream of tokens that are consumed by the Parser.

#### ****2. Token Class (****Token.cs****)****

* **Purpose:** Represents individual tokens.
* **Sequence:**
  + Each token contains information about its type (e.g., keyword, identifier, operator) and value (e.g., int, a, 3).

#### ****3. Parser Class (****Parser.cs****)****

* **Purpose:** Parses the token stream into an Abstract Syntax Tree (AST).
* **Sequence:**
  + Consumes tokens produced by the Lexer.
  + Constructs nodes of different types (e.g., AssignmentNode, ForStatementNode).
  + Validates syntax and reports syntax errors if any.
  + Produces an AST, which is a hierarchical representation of the program.

#### ****4. Abstract Syntax Tree Nodes (AST Nodes)****

Each node represents a specific construct in the program. These include:

**Expression Nodes (**ExpressionNode.cs**)**

* + Handles arithmetic and logical expressions.
  + Examples: AdditionNode, SubtractionNode, MultiplicationNode, etc.

**Statement Nodes:**

* + **AssignmentNode (**AssignmentNode.cs**):**
    - Represents variable assignments (e.g., a = 3;).
  + **ForStatementNode (**ForStatementNode.cs**):**
    - Represents loops (e.g., for a=3 to 5).
  + **IfNodeStatement (**IfNodeStatement.cs**):**
    - Represents conditional statements (e.g., if a > b then ...).

#### ****5. Symbol Table (****SymbolTable.cs****)****

* **Purpose:** Stores information about declared variables and their types.
* **Sequence:**
  + Tracks declared variables, their types, and scopes.
  + Ensures variables are declared before use.
  + Used during both semantic analysis and interpretation.

#### ****6. Semantic Analysis****

* **Classes Involved:**
  + SymbolTable.cs
  + AST nodes (e.g., AssignmentNode, ForStatementNode).
* **Purpose:** Validates the program's semantic correctness.
* **Sequence:**
  + Checks type compatibility (e.g., int cannot be assigned to string).
  + Ensures all variables are declared before use.
  + Throws SemanticException if errors are found.

#### ****7. Interpreter****

* **Classes Involved:**
  + AST nodes (e.g., AssignmentNode, ForStatementNode).
* **Purpose:** Executes the parsed and validated AST.
* **Sequence:**
  + Executes statements in the order they appear in the AST.
  + Evaluates expressions and updates variable values in the symbol table.
  + Produces outputs for print statements.