Logo, company name

Description automatically generated

**COMSATS University Islamabad (CUI)**

**Lab Terminal**

**For**

**CC**

***By***

**Iqra Basharat(FA20-BCS-007)**

***Submitted To:* Sir Bilal**

***Bachelor of Science in Computer Science (2020-2024)***

**Question:01**

***Brief of the project.***

Here's a brief overview of the project:

**1. Transition Table Loading:**

- The loadTransitionTable method reads a transition table from a file specified by the path parameter. This transition table appears to define the transitions between different states based on input characters.

**2. State Transitions:**

- The getNextState method determines the next state in the lexical analysis process based on the current state (iState) and the current input character (cChar).

**3. Token Recognition:**

- The Result method is the main entry point for the lexical analysis.

- It iterates through each character in the input text, updating the state and building tokens.

- Different cases handle the recognition of keywords, identifiers (<ID>), integer literals (<INT>), floating-point literals (<FLOAT>), and string literals (<STR>).

- Operators are recognized as <OPR>.

**4. Comments Handling:**

- The code includes a section for filtering out comments, both single-line (//) and multi-line (/\* \*/).

5. **Keyword Checking:**

- The isKeyword method checks if a given token is a keyword by comparing it to a predefined list of C# keywords.

6. **Output:**

- The result of the lexical analysis is a modified version of the input text where keywords, identifiers, literals, and operators are replaced with their corresponding tokens.

7**. Error Handling:**

- The code includes some basic error handling, such as catching exceptions when reading the transition table file.

**8. Usage:**

- The lexical analyzer is intended to be used by creating an instance of the Analyze class and calling the Result method with the input text.