1. Logic Description

* The program reads lines of code (or user input) using a Scanner and processes them one by one.
* For each line, it invokes the isComment method, which takes a line of code as input and returns true if the line is a comment, following the criteria:
* The isComment method first trims leading and trailing whitespace to remove indentation and checks if the line is empty. An empty line is considered not a comment.
* The line is tokenized into individual tokens using ***line.split("\\s+")***, where **"\\s+"** is a regular expression that splits the line by whitespace.
* If the initial token (the first word) starts with //, the method considers it a single-line comment.
* If the initial token starts with /\*, and the last token (the last word) ends with \*/, the method considers it a multi-line comment.
* Otherwise, it considers the line not to be a comment.
* The main method then prints whether each line is a comment or not based on the result returned by isComment.

1. Were lexical analysis or/and syntax analysis concepts important in your logic? If yes, how?

**Lexical analysis** is important in the logic because the input line is tokenized into individual tokens or lexemes. Tokenization allows the program to examine the first and last tokens independently to identify comment makers (‘//’ and ‘/\*) as part of lexical analysis.

**Syntax analysis** involves parsing and checking the of code structure is not performed in the logic. The program only relies on lexical analysis to categorize lines as comments or non-comments.