Assignment 1

1. Grammar Definition for cal Language

- **Rules for Programs and Declarations**:
    - The main structure of a cal program consists of a decl_list (declaration list), function_list, and a MAIN section.
    - Declarations include variables (var_decl) and constants (const_decl), where each variable and constant has a type (like INT, BOOL, or VOID) and may be initialised by an expression.
- **Statements**:
    - The language supports assignments, function calls, blocks, conditional statements (IF/ELSE), loops (WHILE), and the SKIPS keyword.
- **Expressions and Operators**:
    - The grammar also provides rules for binary arithmetic (PLUS, MINUS) and comparison operators (EQ, NEQ, LT).
- **Lexer Rules for Tokens**:
    - The lexer specifies the reserved words, symbols, and fragments (character sets for case insensitivity), as well as token rules for identifiers, numbers, whitespace, and comments.
    - We cannot rely on the case insensitivity option so to make sure the grammar was case insensitive for each of the alphabet fragments I give the option of upper and lowercase letters.
    - In the grammar, reserved words such as ‘fragment’ and ‘SKIP’ have been changed to ‘fragments’ and ‘SKIPS’ to avoid conflict.
    -

2. Java Parser Class

The main Java class is responsible for reading input, processing it, and identifying any syntax errors.

- **Error Detection**:
    - The parser removes the default error listeners and adds a custom SyntaxErrorListener. This listener sets a flag whenever a syntax error is detected, enabling the main program to check for syntax errors in the input.
- **Output Results**:
    - If the listener detects syntax errors, it reports that the file has not parsed correctly. Otherwise, it confirms successful parsing.

3. Custom Error Listener

- The SyntaxErrorListener class extends BaseErrorListener from ANTLR and overrides the syntaxError method to set a hasErrors flag whenever a syntax error is found. This allows the main program to report errors in a more controlled way.