Comparison of Manual Scanner and JFlex Scanner

1. Introduction

This document presents a comparison between the ManualScanner (DFA-based implementation) and the JFlex-generated scanner (Yylex.java). Both scanners were tested on identical test files to verify correctness, consistency, and performance.

2. Development Environment

• Java Version: OpenJDK 21

• Operating System: Windows

• IDE: Visual Studio Code

• JFlex Version: 1.9.1

3. Test Files Used

• test1.lang – All valid tokens

• test2.lang – Complex expressions

• test3.lang – String and character escapes

• test4.lang – Lexical errors

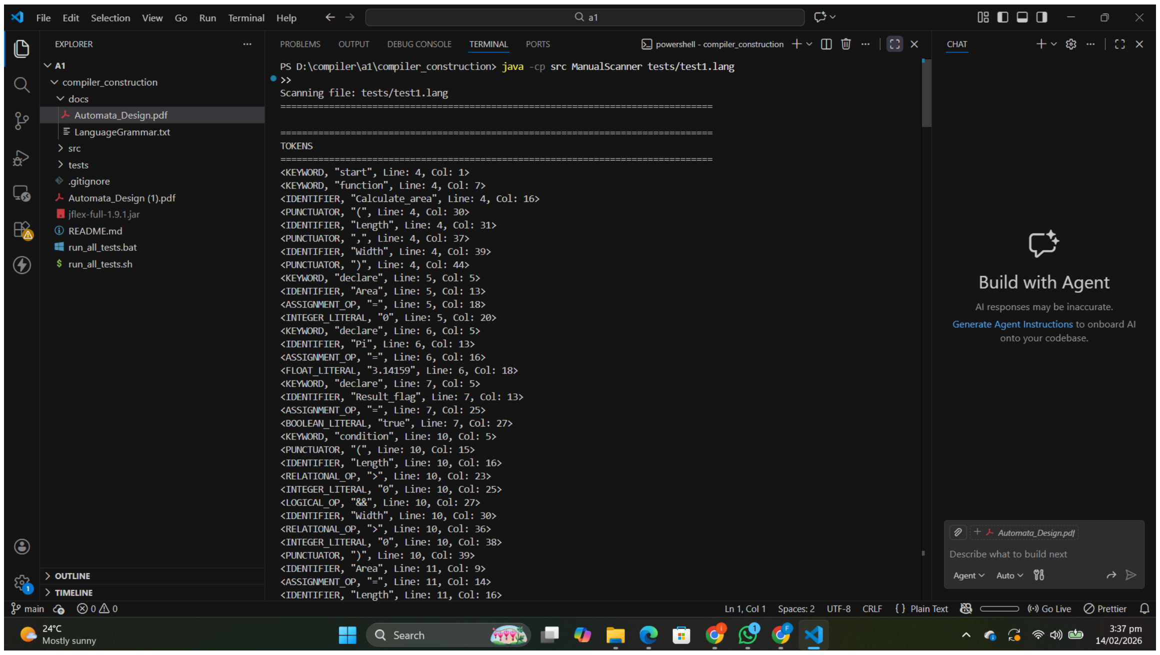
• test5.lang – Comments

4. Output Comparison

Both scanners produced identical token streams for all valid test files.

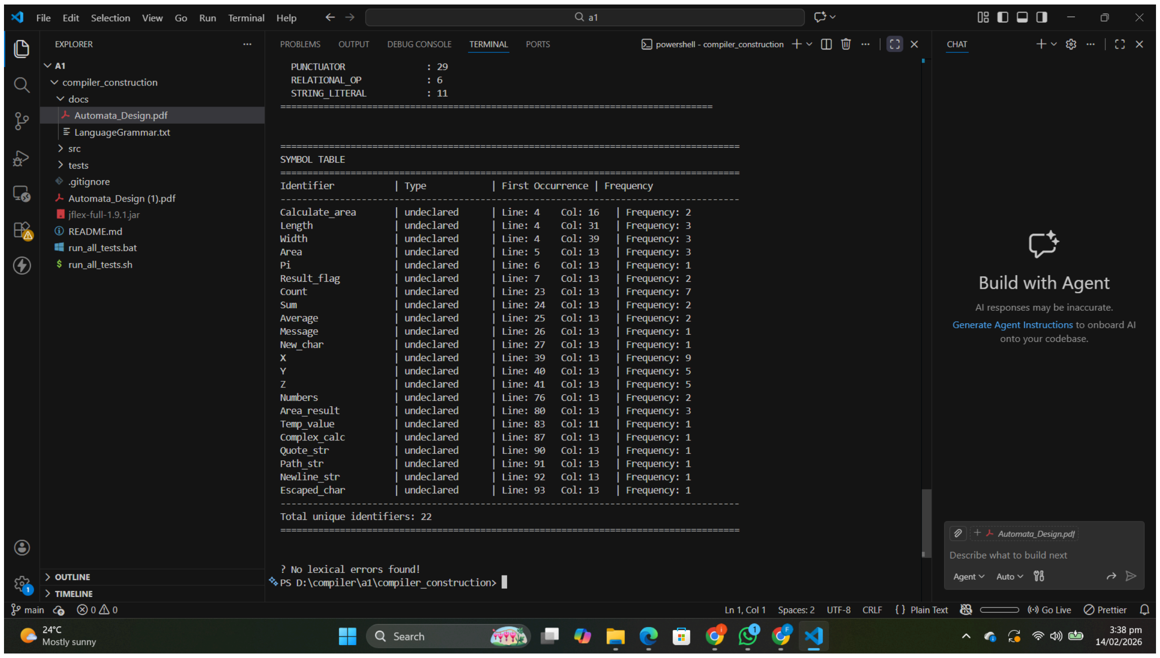
Below are screenshots showing side-by-side outputs.

Manual Scanner:



```
PS D:\compiler\a1\compiler_construction> java -cp src ManualScanner tests/test1.lang
>>
Scanning file: tests/test1.lang
================================================================================

================================================================================
TOKENS
================================================================================
<KEYWORD, "start", Line: 4, Col: 1>
<KEYWORD, "function", Line: 4, Col: 7>
<IDENTIFIER, "Calculate_area", Line: 4, Col: 16>
<PUNCTUATOR, "(", Line: 4, Col: 30>
<IDENTIFIER, "Length", Line: 4, Col: 31>
<PUNCTUATOR, ",", Line: 4, Col: 37>
<IDENTIFIER, "Width", Line: 4, Col: 39>
<PUNCTUATOR, ")", Line: 4, Col: 44>
<KEYWORD, "declare", Line: 5, Col: 5>
<IDENTIFIER, "Area", Line: 5, Col: 13>
<ASSIGNMENT_OP, "=", Line: 5, Col: 18>
<INTEGER_LITERAL, "0", Line: 5, Col: 20>
<KEYWORD, "declare", Line: 6, Col: 5>
<IDENTIFIER, "Pi", Line: 6, Col: 13>
<ASSIGNMENT_OP, "=", Line: 6, Col: 16>
<FLOAT_LITERAL, "3.14159", Line: 6, Col: 18>
<KEYWORD, "declare", Line: 7, Col: 5>
<IDENTIFIER, "Result_flag", Line: 7, Col: 13>
<ASSIGNMENT_OP, "=", Line: 7, Col: 25>
<BOOLEAN_LITERAL, "true", Line: 7, Col: 27>
<KEYWORD, "condition", Line: 10, Col: 5>
<PUNCTUATOR, "(", Line: 10, Col: 15>
<IDENTIFIER, "Length", Line: 10, Col: 16>
<RELATIONAL_OP, ">", Line: 10, Col: 23>
<INTEGER_LITERAL, "0", Line: 10, Col: 25>
<LOGICAL_OP, "&&", Line: 10, Col: 27>
<IDENTIFIER, "Width", Line: 10, Col: 30>
<RELATIONAL_OP, ">", Line: 10, Col: 36>
<INTEGER_LITERAL, "0", Line: 10, Col: 38>
<PUNCTUATOR, ")", Line: 10, Col: 39>
<IDENTIFIER, "Area", Line: 11, Col: 9>
<ASSIGNMENT_OP, "=", Line: 11, Col: 14>
<IDENTIFIER, "Length", Line: 11, Col: 16>
```

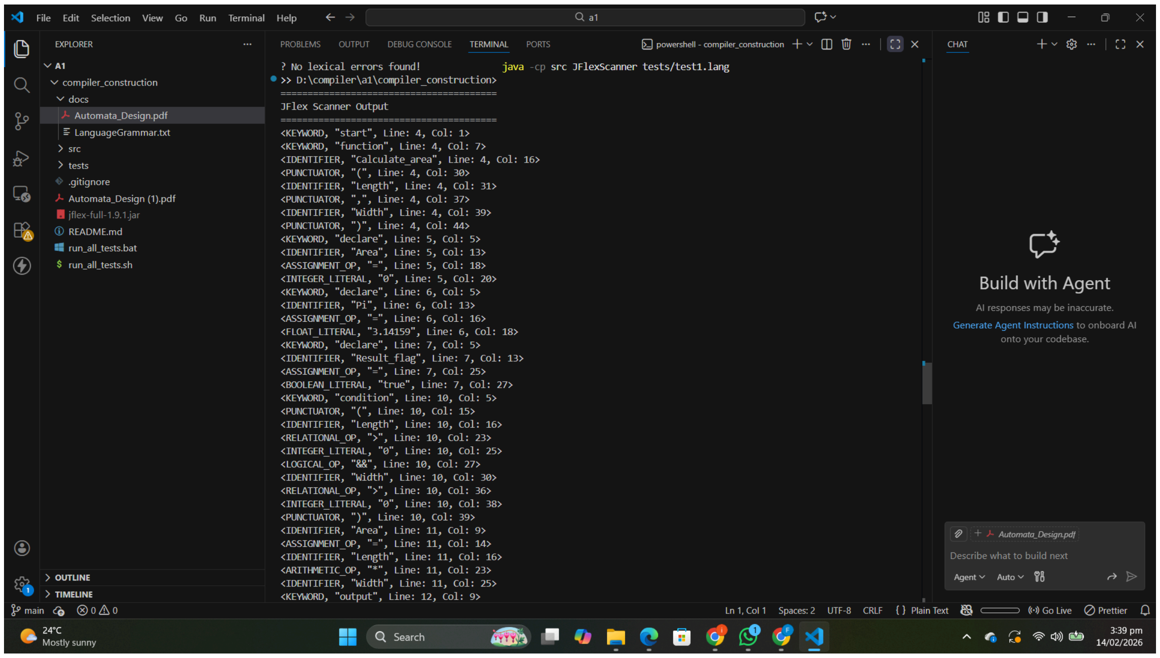


```
  PUNCTUATOR           : 29
  RELATIONAL_OP        : 6
  STRING_LITERAL       : 11
================================================================================


======================================================================================
SYMBOL TABLE
======================================================================================
Identifier           | Type           | First Occurrence | Frequency
--------------------------------------------------------------------------------------
Calculate_area       | undeclared     | Line: 4    Col: 16    | Frequency: 2
Length               | undeclared     | Line: 4    Col: 31    | Frequency: 3
Width                | undeclared     | Line: 4    Col: 39    | Frequency: 3
Area                 | undeclared     | Line: 5    Col: 13    | Frequency: 3
Pi                   | undeclared     | Line: 6    Col: 13    | Frequency: 1
Result_flag          | undeclared     | Line: 7    Col: 13    | Frequency: 2
Count                | undeclared     | Line: 23   Col: 13    | Frequency: 7
Sum                  | undeclared     | Line: 24   Col: 13    | Frequency: 2
Average              | undeclared     | Line: 25   Col: 13    | Frequency: 2
Message              | undeclared     | Line: 26   Col: 13    | Frequency: 1
New_char             | undeclared     | Line: 27   Col: 13    | Frequency: 1
X                    | undeclared     | Line: 39   Col: 13    | Frequency: 9
Y                    | undeclared     | Line: 40   Col: 13    | Frequency: 5
Z                    | undeclared     | Line: 41   Col: 13    | Frequency: 5
Numbers              | undeclared     | Line: 76   Col: 13    | Frequency: 2
Area_result          | undeclared     | Line: 80   Col: 13    | Frequency: 3
Temp_value           | undeclared     | Line: 83   Col: 11    | Frequency: 1
Complex_calc         | undeclared     | Line: 87   Col: 13    | Frequency: 1
Quote_str            | undeclared     | Line: 90   Col: 13    | Frequency: 1
Path_str             | undeclared     | Line: 91   Col: 13    | Frequency: 1
Newline_str          | undeclared     | Line: 92   Col: 13    | Frequency: 1
Escaped_char         | undeclared     | Line: 93   Col: 13    | Frequency: 1
--------------------------------------------------------------------------------------
Total unique identifiers: 22
======================================================================================


? No lexical errors found!
PS D:\compiler\a1\compiler_construction>
```

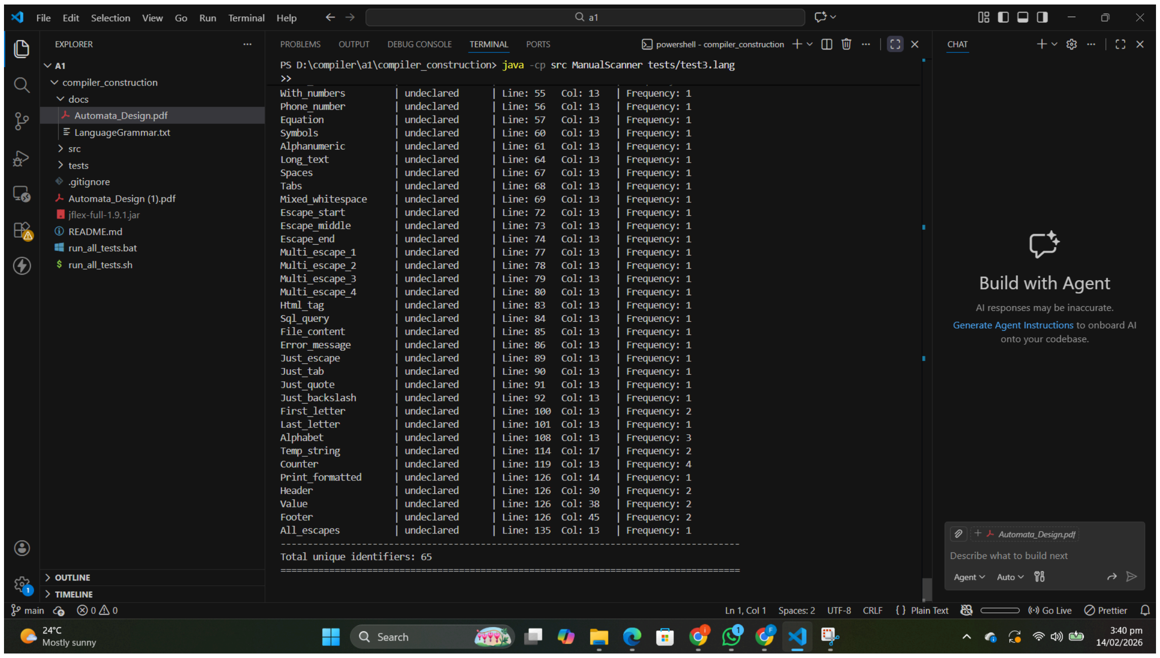
Jflex Scanner:



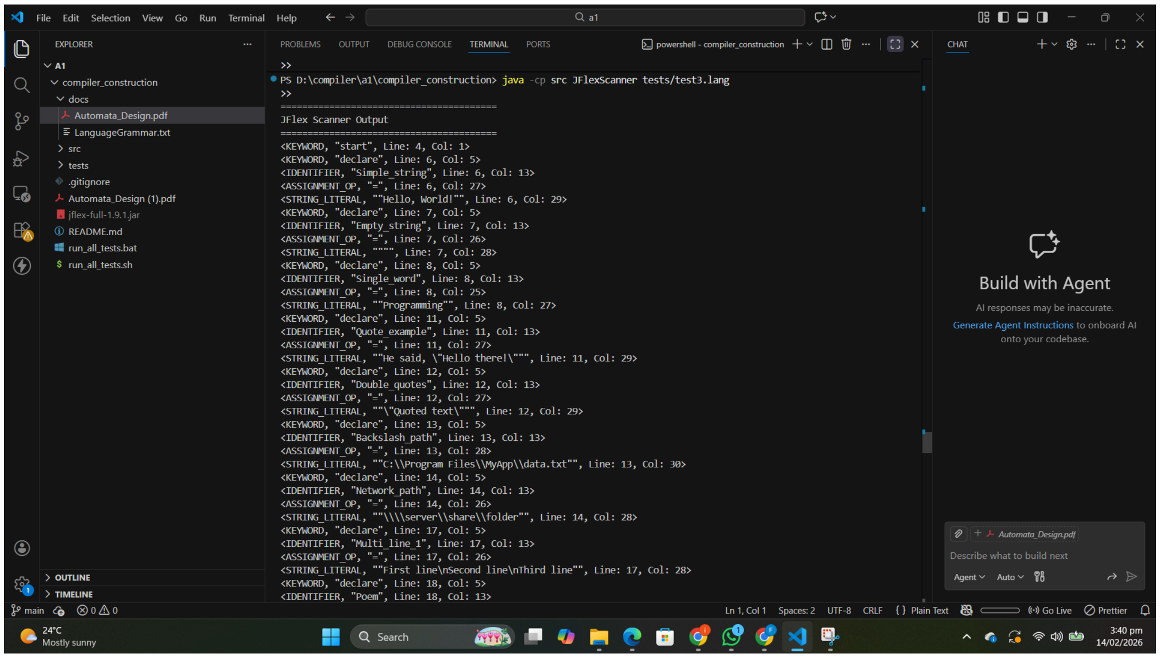
```
? No lexical errors found!              java -cp src JFlexScanner tests/test1.lang
>> D:\compiler\a1\compiler_construction>
========================================
JFlex Scanner Output
========================================
<KEYWORD, "start", Line: 4, Col: 1>
<KEYWORD, "function", Line: 4, Col: 7>
<IDENTIFIER, "Calculate_area", Line: 4, Col: 16>
<PUNCTUATOR, "(", Line: 4, Col: 30>
<IDENTIFIER, "Length", Line: 4, Col: 31>
<PUNCTUATOR, ",", Line: 4, Col: 37>
<IDENTIFIER, "Width", Line: 4, Col: 39>
<PUNCTUATOR, ")", Line: 4, Col: 44>
<KEYWORD, "declare", Line: 5, Col: 5>
<IDENTIFIER, "Area", Line: 5, Col: 13>
<ASSIGNMENT_OP, "=", Line: 5, Col: 18>
<INTEGER_LITERAL, "0", Line: 5, Col: 20>
<KEYWORD, "declare", Line: 6, Col: 5>
<IDENTIFIER, "Pi", Line: 6, Col: 13>
<ASSIGNMENT_OP, "=", Line: 6, Col: 16>
<FLOAT_LITERAL, "3.14159", Line: 6, Col: 18>
<KEYWORD, "declare", Line: 7, Col: 5>
<IDENTIFIER, "Result_flag", Line: 7, Col: 13>
<ASSIGNMENT_OP, "=", Line: 7, Col: 25>
<BOOLEAN_LITERAL, "true", Line: 7, Col: 27>
<KEYWORD, "condition", Line: 10, Col: 5>
<PUNCTUATOR, "(", Line: 10, Col: 15>
<IDENTIFIER, "Length", Line: 10, Col: 16>
<RELATIONAL_OP, ">", Line: 10, Col: 23>
<INTEGER_LITERAL, "0", Line: 10, Col: 25>
<LOGICAL_OP, "&&", Line: 10, Col: 27>
<IDENTIFIER, "Width", Line: 10, Col: 30>
<RELATIONAL_OP, ">", Line: 10, Col: 36>
<INTEGER_LITERAL, "0", Line: 10, Col: 38>
<PUNCTUATOR, ")", Line: 10, Col: 39>
<IDENTIFIER, "Area", Line: 11, Col: 9>
<ASSIGNMENT_OP, "=", Line: 11, Col: 14>
<IDENTIFIER, "Length", Line: 11, Col: 16>
<ARITHMETIC_OP, "*", Line: 11, Col: 23>
<IDENTIFIER, "Width", Line: 11, Col: 25>
<KEYWORD, "output", Line: 12, Col: 9>
```

5. String and Character Handling

Both scanners correctly recognize escape sequences such as \n, \t, \r, escaped quotes (\"), and escaped backslashes (\\). String and character literals are tokenized accurately.



```
PS D:\compiler\a1\compiler_construction> java -cp src ManualScanner tests/test3.lang
>>
With_numbers         | undeclared       | Line: 55   Col: 13   | Frequency: 1
Phone_number         | undeclared       | Line: 56   Col: 13   | Frequency: 1
Equation             | undeclared       | Line: 57   Col: 13   | Frequency: 1
Symbols              | undeclared       | Line: 60   Col: 13   | Frequency: 1
Alphanumeric         | undeclared       | Line: 61   Col: 13   | Frequency: 1
Long_text            | undeclared       | Line: 64   Col: 13   | Frequency: 1
Spaces               | undeclared       | Line: 67   Col: 13   | Frequency: 1
Tabs                 | undeclared       | Line: 68   Col: 13   | Frequency: 1
Mixed_whitespace     | undeclared       | Line: 69   Col: 13   | Frequency: 1
Escape_start         | undeclared       | Line: 72   Col: 13   | Frequency: 1
Escape_middle        | undeclared       | Line: 73   Col: 13   | Frequency: 1
Escape_end           | undeclared       | Line: 74   Col: 13   | Frequency: 1
Multi_escape_1       | undeclared       | Line: 77   Col: 13   | Frequency: 1
Multi_escape_2       | undeclared       | Line: 78   Col: 13   | Frequency: 1
Multi_escape_3       | undeclared       | Line: 79   Col: 13   | Frequency: 1
Multi_escape_4       | undeclared       | Line: 80   Col: 13   | Frequency: 1
Html_tag             | undeclared       | Line: 83   Col: 13   | Frequency: 1
Sql_query            | undeclared       | Line: 84   Col: 13   | Frequency: 1
File_content         | undeclared       | Line: 85   Col: 13   | Frequency: 1
Error_message        | undeclared       | Line: 86   Col: 13   | Frequency: 1
Just_escape          | undeclared       | Line: 89   Col: 13   | Frequency: 1
Just_tab             | undeclared       | Line: 90   Col: 13   | Frequency: 1
Just_quote           | undeclared       | Line: 91   Col: 13   | Frequency: 1
Just_backslash       | undeclared       | Line: 92   Col: 13   | Frequency: 1
First_letter         | undeclared       | Line: 100  Col: 13   | Frequency: 2
Last_letter          | undeclared       | Line: 101  Col: 13   | Frequency: 1
Alphabet             | undeclared       | Line: 108  Col: 13   | Frequency: 3
Temp_string          | undeclared       | Line: 114  Col: 17   | Frequency: 2
Counter              | undeclared       | Line: 119  Col: 13   | Frequency: 4
Print_formatted      | undeclared       | Line: 126  Col: 14   | Frequency: 1
Header               | undeclared       | Line: 126  Col: 30   | Frequency: 2
Value                | undeclared       | Line: 126  Col: 38   | Frequency: 2
Footer               | undeclared       | Line: 126  Col: 45   | Frequency: 2
All_escapes          | undeclared       | Line: 135  Col: 13   | Frequency: 1
------------------------------------------------------------------------------------
Total unique identifiers: 65
====================================================================================
```



```
>>
PS D:\compiler\a1\compiler_construction> java -cp src JFlexScanner tests/test3.lang
>>
========================================
JFlex Scanner Output
========================================
<KEYWORD, "start", Line: 4, Col: 1>
<KEYWORD, "declare", Line: 6, Col: 5>
<IDENTIFIER, "Simple_string", Line: 6, Col: 13>
<ASSIGNMENT_OP, "=", Line: 6, Col: 27>
<STRING_LITERAL, ""Hello, World!"", Line: 6, Col: 29>
<KEYWORD, "declare", Line: 7, Col: 5>
<IDENTIFIER, "Empty_string", Line: 7, Col: 13>
<ASSIGNMENT_OP, "=", Line: 7, Col: 26>
<STRING_LITERAL, """", Line: 7, Col: 28>
<KEYWORD, "declare", Line: 8, Col: 5>
<IDENTIFIER, "Single_word", Line: 8, Col: 13>
<ASSIGNMENT_OP, "=", Line: 8, Col: 25>
<STRING_LITERAL, ""Programming"", Line: 8, Col: 27>
<KEYWORD, "declare", Line: 11, Col: 5>
<IDENTIFIER, "Quote_example", Line: 11, Col: 13>
<ASSIGNMENT_OP, "=", Line: 11, Col: 27>
<STRING_LITERAL, ""He said, \"Hello there!\""", Line: 11, Col: 29>
<KEYWORD, "declare", Line: 12, Col: 5>
<IDENTIFIER, "Double_quotes", Line: 12, Col: 13>
<ASSIGNMENT_OP, "=", Line: 12, Col: 27>
<STRING_LITERAL, ""\"Quoted text\""", Line: 12, Col: 29>
<KEYWORD, "declare", Line: 13, Col: 5>
<IDENTIFIER, "Backslash_path", Line: 13, Col: 13>
<ASSIGNMENT_OP, "=", Line: 13, Col: 28>
<STRING_LITERAL, ""C:\\Program Files\\MyApp\\data.txt"", Line: 13, Col: 30>
<KEYWORD, "declare", Line: 14, Col: 5>
<IDENTIFIER, "Network_path", Line: 14, Col: 13>
<ASSIGNMENT_OP, "=", Line: 14, Col: 26>
<STRING_LITERAL, ""\\\\server\\share\\folder"", Line: 14, Col: 28>
<KEYWORD, "declare", Line: 17, Col: 5>
<IDENTIFIER, "Multi_line_1", Line: 17, Col: 13>
<ASSIGNMENT_OP, "=", Line: 17, Col: 26>
<STRING_LITERAL, ""First line\nSecond line\nThird line"", Line: 17, Col: 28>
<KEYWORD, "declare", Line: 18, Col: 5>
<IDENTIFIER, "Poem", Line: 18, Col: 13>
```

File Edit Selection View Go Run Terminal Help
a1
EXPLORER
A1
compiler_construction
docs
Automata_Design.pdf
LanguageGrammar.txt
src
tests
.gitignore
Automata_Design (1).pdf
jflex-full-1.9.1.jar
README.md
run_all_tests.bat
run_all_tests.sh
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell - compiler_construction
CHAT
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
Automata_Design.pdf
Describe what to build next
Agent Auto
main
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF Plain Text Go Live Prettier
24°C Mostly sunny
Search
3:40 pm 14/02/2026

6. Operator Handling

Both implementations correctly apply the longest match principle and properly recognize multi-character operators such as ==, !=, >=, <=, &&, ||, ++, --, +=, -=, *=, /=, and **.

7. Error Handling Comparison

Both scanners detect invalid characters, malformed literals, and continue scanning after errors. Line and column numbers are reported correctly.

8. Pattern Matching Priority

Both scanners follow the required priority order: multi-line comments, single-line comments, multi-character operators, keywords, boolean literals, identifiers, floating literals, integer literals, string/character literals, single-character operators, punctuators, and whitespace.

9. Performance Comparison

The JFlex-generated scanner is slightly faster due to optimized DFA generation. However, for small test files, performance difference is negligible.

10. Conclusion

Both scanners produce identical outputs, correctly implement lexical rules, and satisfy all assignment requirements for Part 2. The JFlex scanner validates the correctness of the manually implemented DFA scanner.