Compilers Theory Assignment

The goal of the assignment is to understand how compilers are implemented in practice. You are given a source code of a simple language interpreter written in Java. Three phases are implemented: lexical analysis, syntactic analysis and semantic analysis. You are also given a modified version of the language with more rules and your task is to add these rules to the interpreter across the three phases.

- **Prerequisite knowledge for the assignment**: (context-free grammar in automata theory, basic concepts in compilers theory, programming basics in Java, object-oriented programming concepts, recursion and visitor design pattern).
- **Study the following grammar through parsing a few expressions by hand then examine the attached code carefully:**

program → expr*
expr → factor ("+" factor)*
factor → unary
unary → number
number → digit^{+} ("." digit*)
digit → 0|1|2|3|4|5|6|7|8|9

  - **Valid expressions:**
    - **5 + 3**
    - **0 + 1.5 + 234**
    - **6**
- **The following is a modified version of the grammar which allows subtraction, division, multiplication and negation:**

program → expr*
expr → factor (("-" | "+") factor)*
factor → factor ("*" | "/") unary | unary
unary → ("-") number
number → digit^{+} ("." digit*)
digit → 0|1|2|3|4|5|6|7|8|9

  - **Valid expressions:**
    - **5 + 3 .5 * 2 - 1**
    - **3 * 10 / 5**
- **A few notes to make sure you understand the grammar correctly:**
  - *** → zero or more times**
  - **+ → one or more times**
  - **(x | y) → x or y**
  - **("-") is optional**
  - **("." digit*) is optional**

Tasks:

1. Update the lexical analyzer to identify these characters ("-", "*", "/") and add them to the list of tokens. Your code for this part will be written inside *"Lexer.java"*, you only need to update *"scanToken()"* function.

2. Update the parser with the new rules. Your code for this part will be written inside *"Parser.java"*.
   a. Modify *"expression()"* function to support parsing "Minus" tokens as well.
   b. Modify *"factor()"* function to support parsing "Slash" and "Star" tokens as well.
   c. Modify *"unary()"* function to support parsing "Minus" (if it exists) before parsing a number.

3. Update the semantic analyzer to semantically analyze and evaluate binary expressions involving these operators: "-", "*", "/". Your code for this part will be written inside *"Interpreter.java"*.
   a. Modify *"visitBinaryExpr()"* function to handle evaluation for these operators "-", "*", "/".
   b. Modify *"visitUnaryExpr()"* function to handle the negation unary operator "-" and evaluate the negated number.

Deliverables:

Your report must include the following:

1. Test your code with an expression that contains all operators (+, -, *, /) and show the input and output in the report.
2. Attach complete source code of "Lexer.java", "Parser.java" and "Interpreter.java" after the updates in the report. **Highlight updated lines in the code.**