**LAB TERMINAL**

**Submitted to: Sir Bilal Haider**

**Submitted by: Sara Ali**

**Reg: Fa21-bcs-011**

**Course: CC**

**Repository of Compiler:**

https://github.com/nrother/simple-c.git

**Ans1:**

**1)Tokenizing string input from tokenizer.cs**

The purpose is to split input source code into tokens**.**

**Example Input:** int x = 5;

**Example Output:**

**A list of tokens:** [Keyword(int), Identifier(x), Operator(=), NumberLiteral(5), StatementSeparator(;)]

**2)Parsing expression from the file Expression.cs**

Expression.cs constructs an AST for the expressions(Abstract binary tree). It translates a small subset of C Like code and processes tokens to generate AST.

**Example Input: x + y \* 2**

**Example Output:**

**BinaryOperationNodes**: It uses recursive descent to handle operator precedance

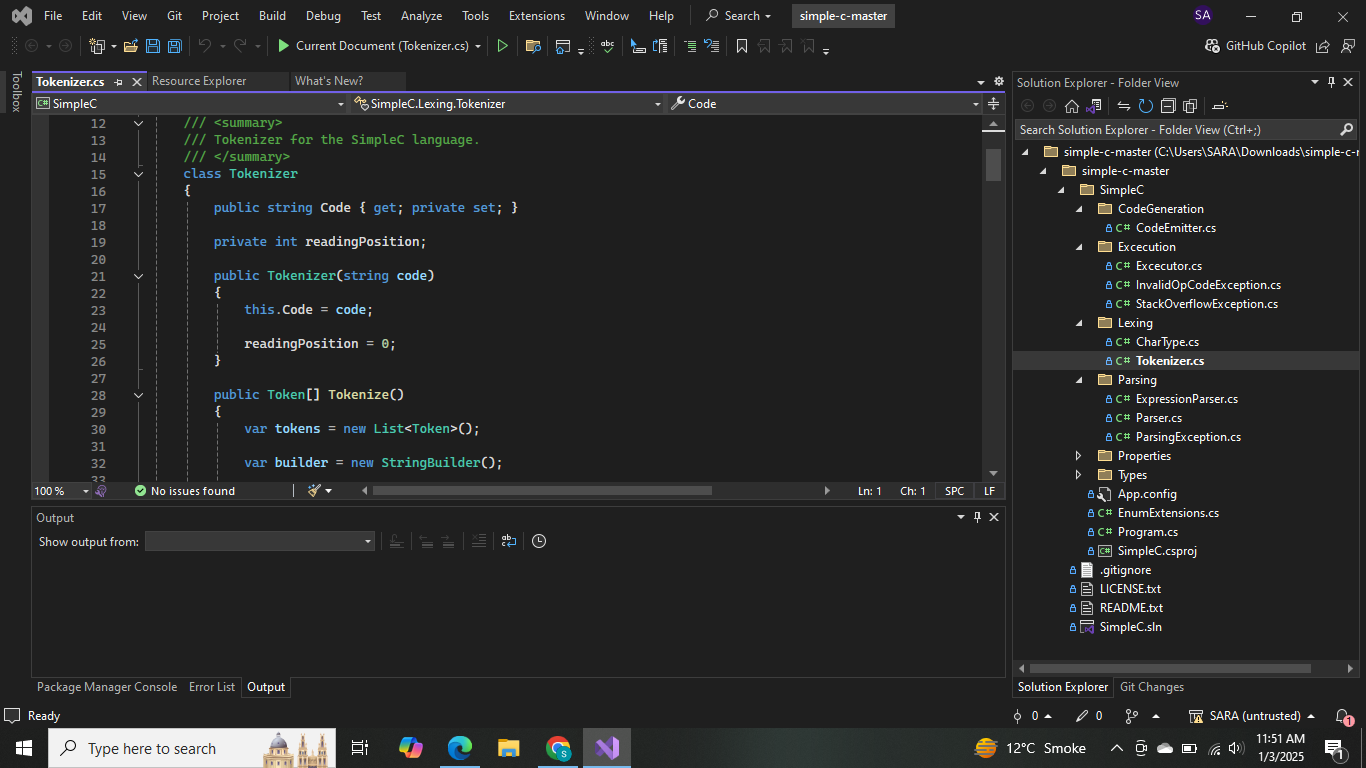
Left: VariableReference(x)

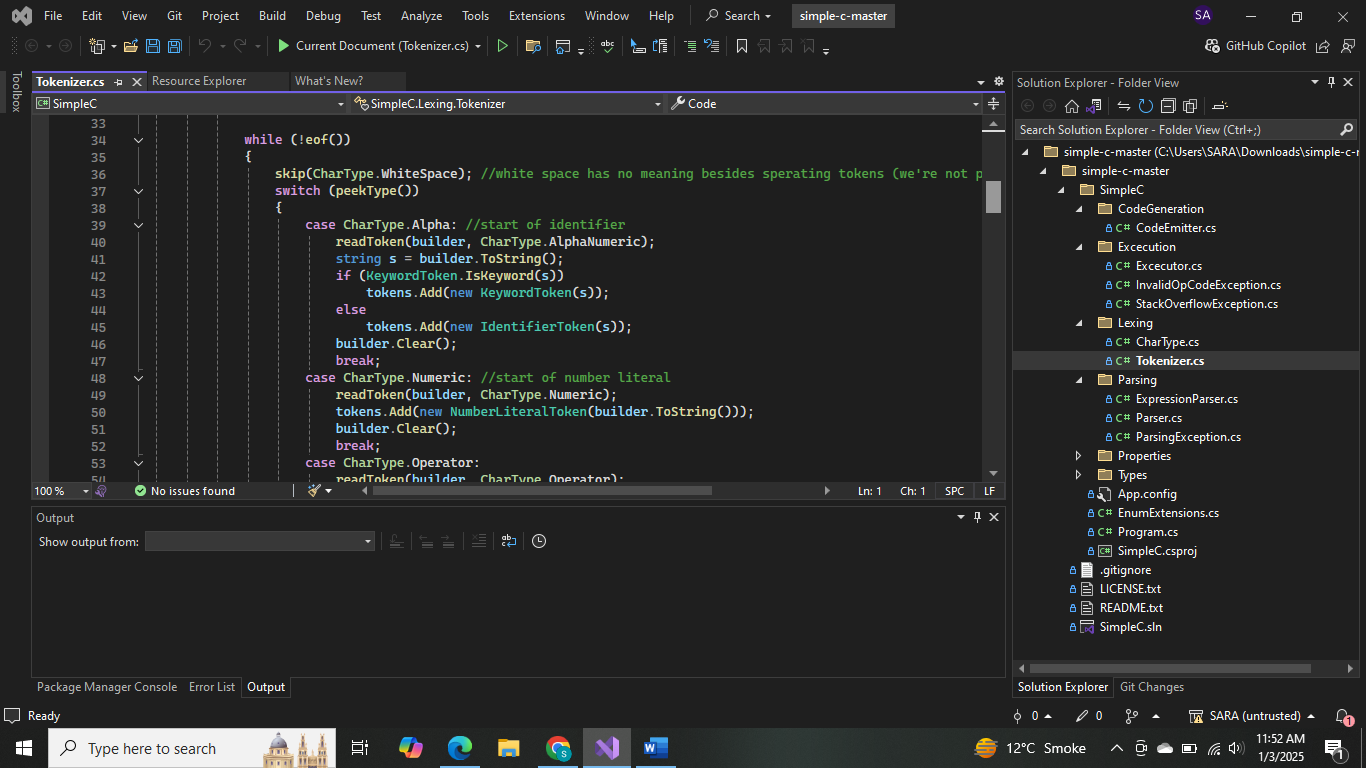
Right: BinaryOperationNode

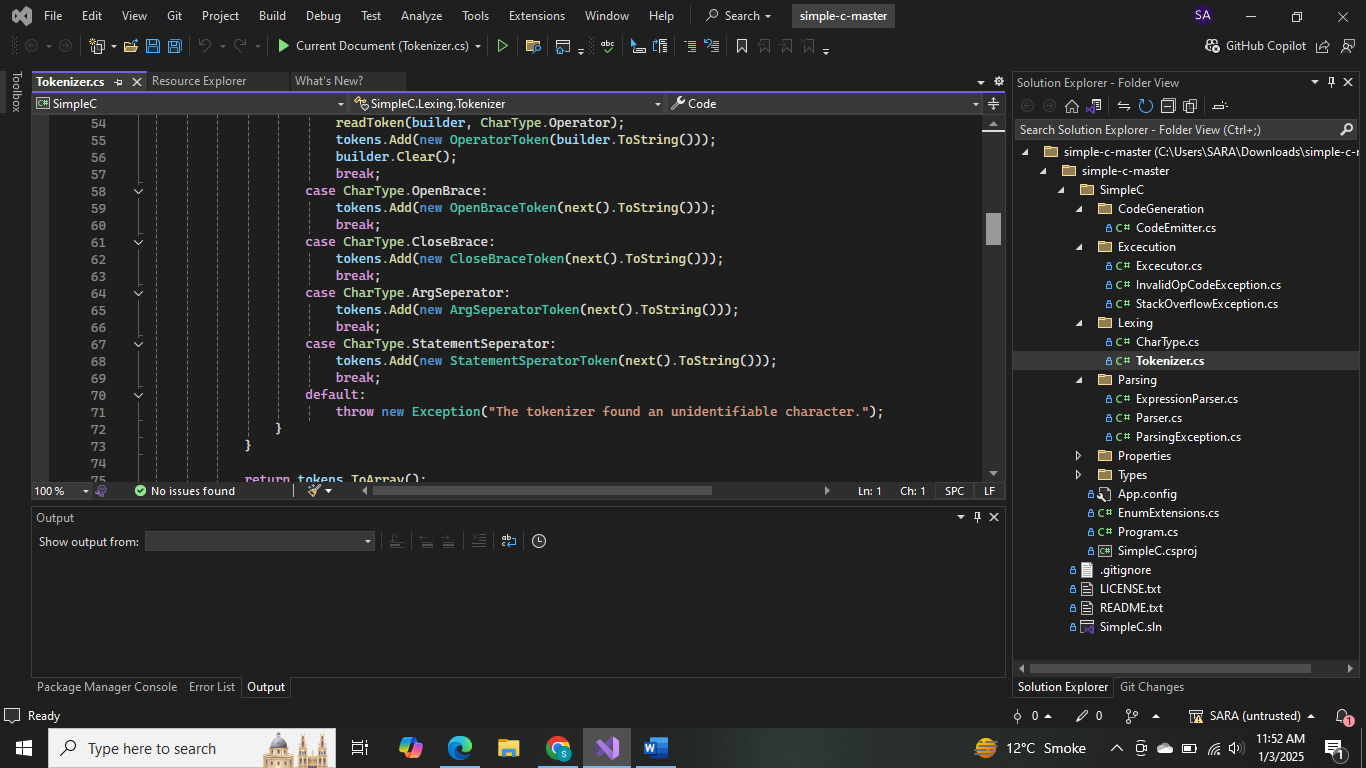
Left: VariableReference(y)

Right: NumberLiteral(2)

**Tokenizer.cs**







**ExpressionParser.cs:** Need to check for the operators

