**LAB TERMINAL**

**Submitted to: Sir Bilal Haider**

**Submitted by: Sara Ali**

**Reg: Fa21-bcs-011**

**Course: CC**

**Ans2) Input and Output of Compiler:**

**Input:**

Input is the source code written in simple c language

**Program.cs**: It is the main entry point for the compiler and basic compilation process. We have declared a Sample c-like program as a string

class Program

{

static void Main(string[] args)

{

string code = @"

int a = 5;

int func(int b)

{

int c = (5\*b)+7;

return c;

}

int main()

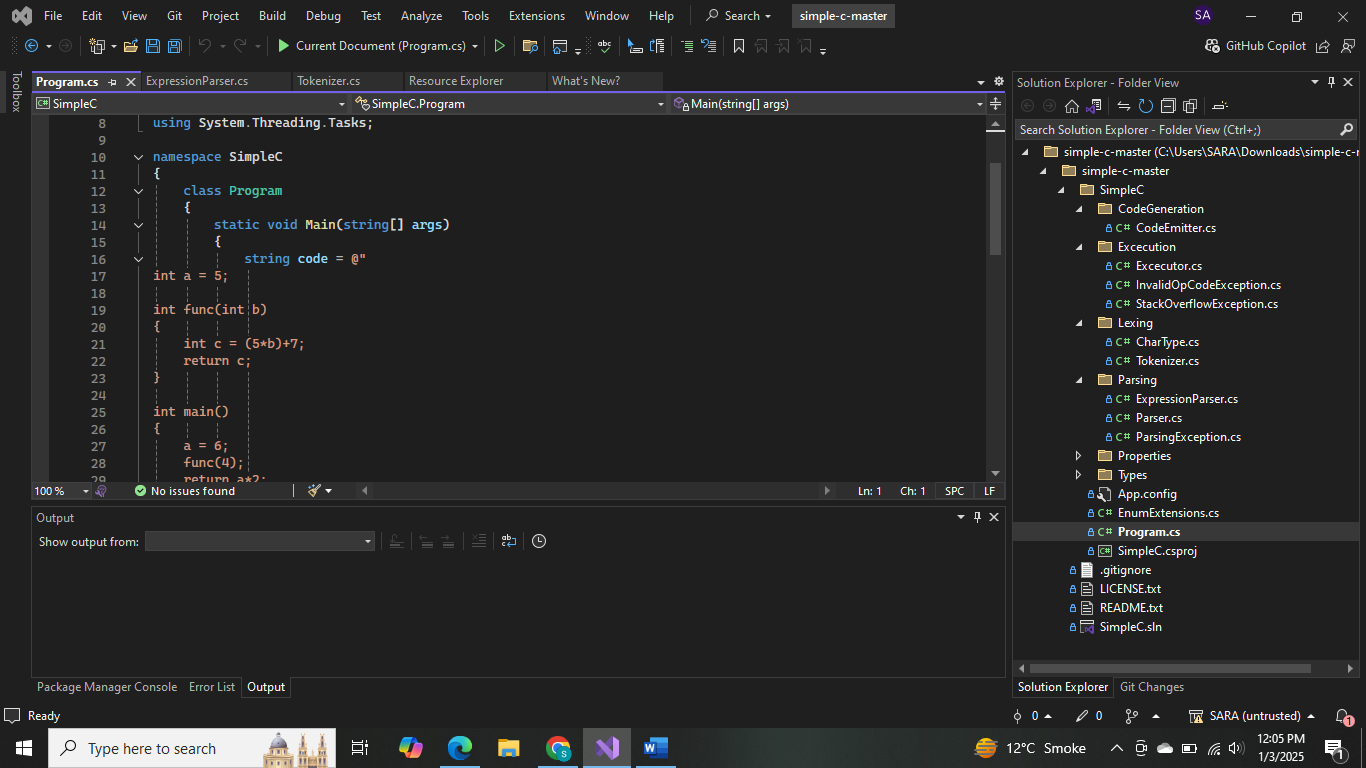
{

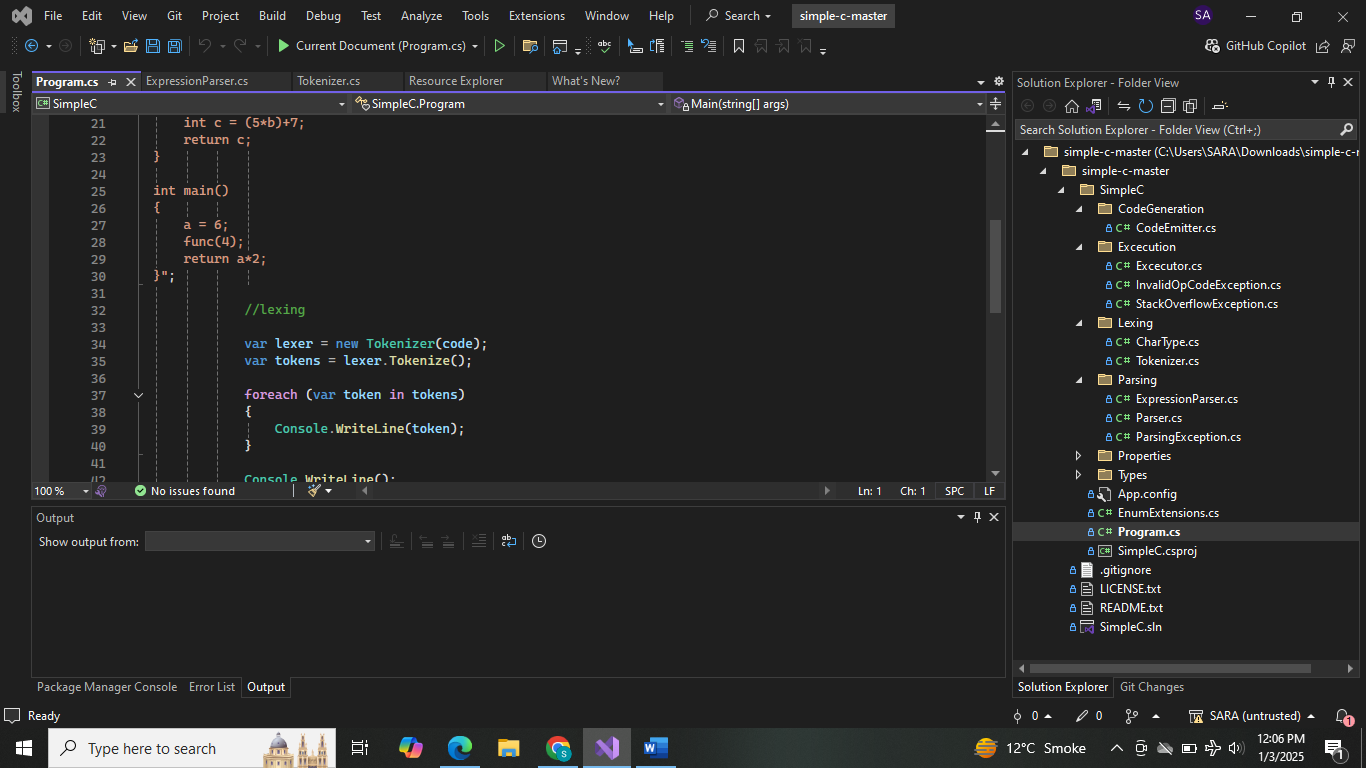
a = 6;

func(4);

return a\*2;

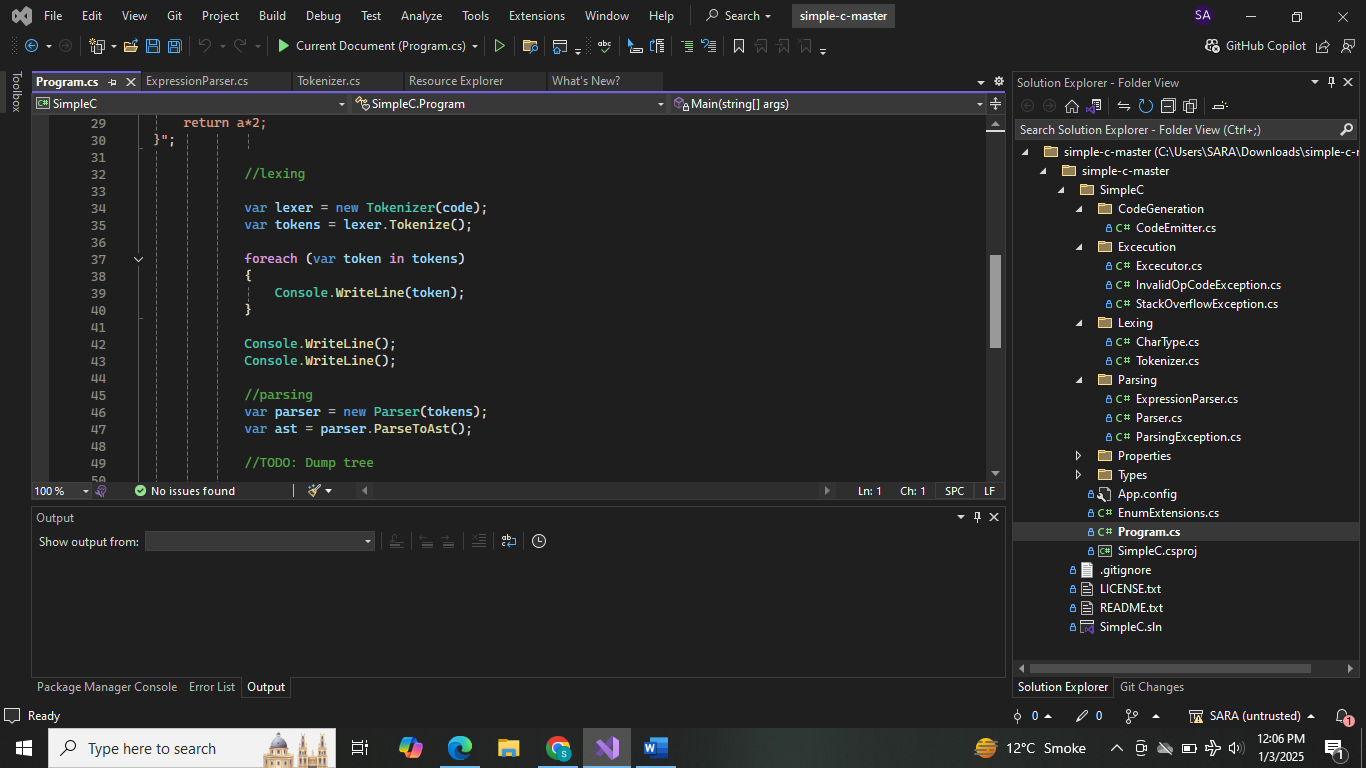
}";





**Lexing:**

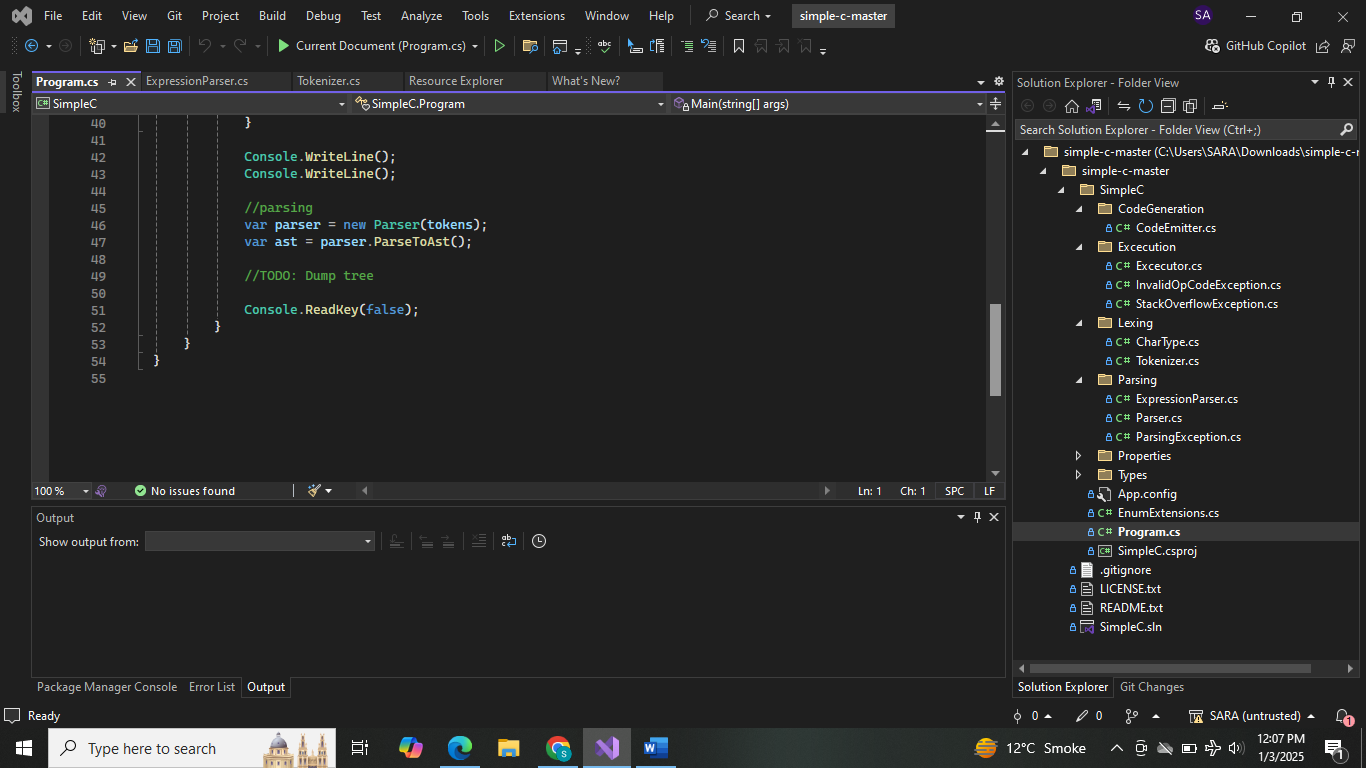
1. Creates tokenizer object with input code string
2. Calls Tokenize to convert the code into tokens
3. Iterates through tokens, printing each



**Parsing:**

After tokenization, the Parser processes the tokens to generate an AST. The AST represents the hirerchical structure of the program and creates relationship between its components.

Example: An expression like a=6; would be represented as an assignment node with a as a target and 6 is a value.



**Output:**

1. Tokens generated during lexical analysis.

2. AST representation from the parser.

3. Intermediate code for the virtual machine.

4. Final output: The result of the interpreted code (e.g., 5 for the above example).

