**Lab Terminal**   
 ****

**Submitted By :Nida Eman**

**Reg No: fa21-bcs-002**

**Submitted to : Sir Bilal Haider**

**Course: Compiler Construction**

**Date: 3/1/2025**

**QS 4:Explain the function of semantic analysis in minicompiler.**

This stage checks the program's logical structure to ensure it adheres to the semantic rules of the programming language, such as verifying type compatibility, scope resolution, and required constructs like return statements.

**How It Fits in the Mini Compiler**

* **Syntax vs. Semantic:**

While syntax analysis ensures the structure of the code is correct (e.g., proper grammar), semantic analysis verifies logical correctness (e.g., presence of a return statement in a non-void function).

* **Error Detection:**

If the source code lacks a required return statement, the Analyze method flags the issue.

**Enhancements**

This semantic analysis could be expanded to include additional checks, such as:

* **Type Checking:**

Verify that return types match the function signature.

* **Scope Resolution:**

Ensure variables and functions are declared before use.

* **Control Flow Validation:**

Ensure all code paths in a function with a non-void return type lead to a return statement.

Example:

List<Tuple<string, string>> tokens = new List<Tuple<string, string>>()

{

Tuple.Create("KEYWORD", "int"),

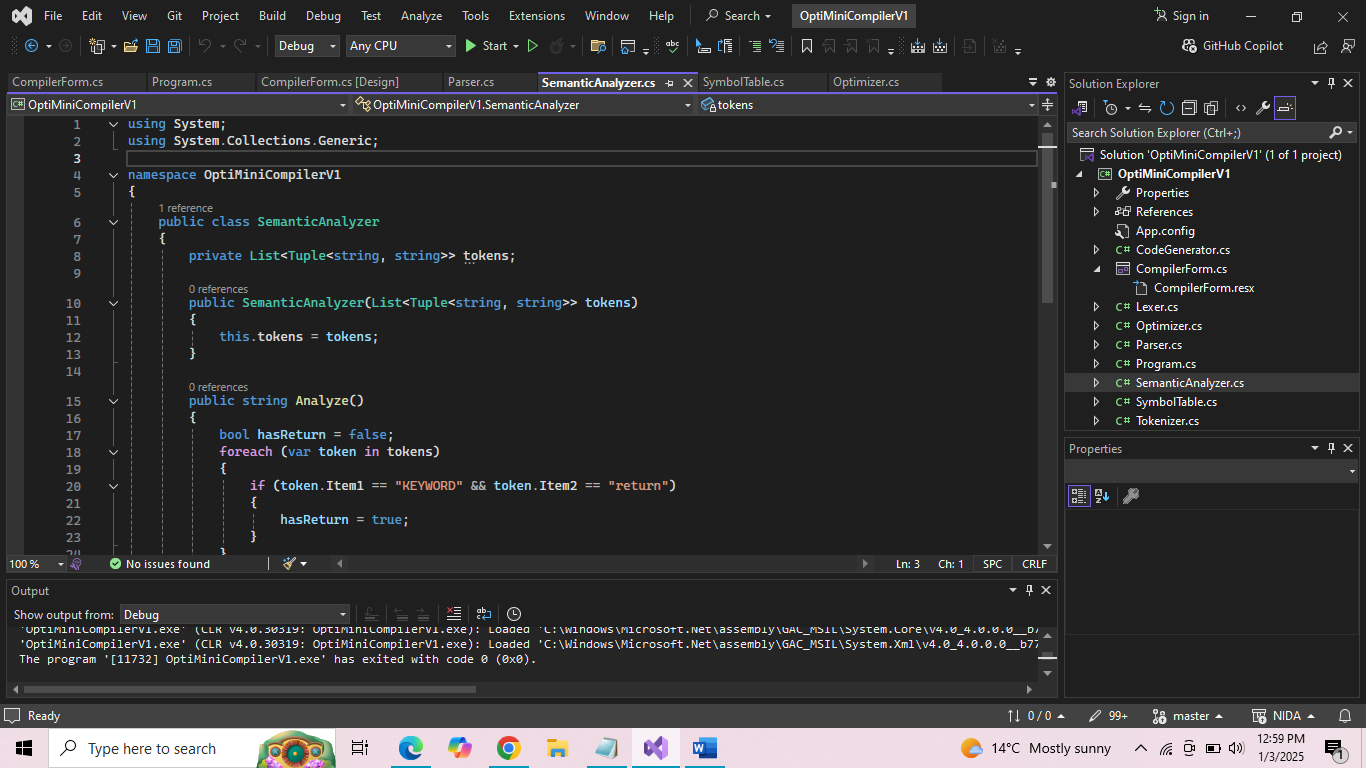
Tuple.Create("IDENTIFIER", "a"),

Tuple.Create("KEYWORD", "return"),

Tuple.Create("IDENTIFIER", "b")

};

Output: Semantic Analysis Passed

**Code:** 

**Analyze method:** 