**Semantic Analysis:**

The semantic analysis phase ensures that operations are type-safe, function calls are valid, and variables are correctly used in their scopes. Here’s a brief breakdown:

**Function and Variable Declaration (FuncStmt, Write):**

The code checks if functions and variables are declared before use by verifying against functionTable and symbolTable.

For function calls, it ensures the number and types of parameters match the function’s definition.

**Operator Handling (SemanticHelper):**

Operators are evaluated with correct precedence using functions like GetOpWeight and HasHigherPre.

It converts expressions from infix to postfix using InfixToPostfix, allowing for proper evaluation of expressions.

**Type Checking:**

In FuncStmt, it validates that parameters passed to functions have the correct data types. If there’s a mismatch, an error is added to the list.

The EvaluateExp and EvaluateFloatExp functions evaluate arithmetic expressions (integer and float) and check for type consistency.

**Scope and Context Validation:**

The scope is tracked during function and statement evaluation, ensuring variables are used within their valid scope (currentScope).