

**Lab Terminal**

**Submitted By:**

Ramsha Kokab

FA21-BCS-010

**Course Instructor:**

Syed Bilal Haider Bukhari

**Course:**

Compiler Construction [Lab]

**Date:**

3rd January, 2024

**DEPARTMENT OF COMPUTER SCIENCE**

**COMSATS UNIVERSITY ISLAMABAD, ATTOCK CAMPUS**

### Q5: Explain the function that performs the semantic analysis in mini compiler.

#### ****Function Name:****

The function responsible for semantic analysis might be named something like *semanticAnalyzer()* or checkSemantics().

void semanticAnalyzer(Node\* syntaxTree, SymbolTable\* symbolTable) {

    // Traverse the syntax tree

    if (syntaxTree == NULL) {

        return;

    }

    // Example 1: Check for undeclared variables

    if (syntaxTree->type == VARIABLE) {

        if (!isDeclared(syntaxTree->value, symbolTable)) {

            printf("Error: Undeclared variable '%s'.\n", syntaxTree->value);

        }

    }

    // Example 2: Type checking for operations

    if (syntaxTree->type == OPERATION) {

        DataType leftType = getNodeType(syntaxTree->left);

        DataType rightType = getNodeType(syntaxTree->right);

        if (leftType != rightType) {

            printf("Error: Type mismatch in operation '%s'.\n", syntaxTree->value);

        }

    }

    // Recursively analyze child nodes

    semanticAnalyzer(syntaxTree->left, symbolTable);

    semanticAnalyzer(syntaxTree->right, symbolTable);

}

* *Output:*

The semantic analyzer outputs:

* An updated symbol table with type information, scope, and variable usage.
* Error messages for semantic issues such as type mismatches, undeclared variables, or scope violations.
* A successful validation message if no semantic errors are found.

