# SEMANTIC ANALYZER

## Q1: Brief Introduction of Project

## ANSWER:

The project involves a basic compiler or interpreter that processes user-entered source code. The main components include a Scanner, Semantic Analyzer.

Semantic Analyzer:

This component applies semantic rules to the tokens generated by the scanner. It checks for semantic errors, ensuring that the code makes logical sense. This includes checking for proper variable declaration and assignment, as well as verifying that operations are performed between compatible types. Following are the rules that my Semantic Analyzer follows:

RULE 1:

First rule checks the datatypes that used in source code (input).

RULE 2:

Second rule checks operators (+, -, *, /) in source code (input).

RULE 3:

Third rule checks logical operators (>=, <=, >, <) in

source code (input).

The SemanticAnalyzer() function takes the list of tokens

identified by the Scanner, and checks if the tokens sequence

follow any of the three rules provided in the project

description using Top-Down Parsing, then it return a list of the

errors found.

Scanner:

The scanner is responsible for breaking down the source

code into individual tokens. It classifies these tokens into

categories such as identifiers, numbers, variables, operators,

etc.