



COMPILER CONSTRUCTION

LAB TERMINAL

NAME:

Muhammad Aamir Bin Habib (FA20-BCS-010)

CLASS:

BCS-7B

Submitted To:

Mr. Bilal Haider

Date:

27-December -2023

QUESTION NO.1

Brief Of Project:

We have developed a mini-compiler that works for a Java source code and has following functionalities:

- **Scanner (Lexical Analysis):**

The scanner is the first phase of the compiler that reads the source code written in Java and breaks it down into tokens. Tokens are the smallest units of the code, such as keywords, identifiers, literals, operators, and punctuation marks. The scanner ensures that the code is well-formed by identifying and categorizing these tokens according to the rules of the Java language. It eliminates comments and whitespace, focusing on extracting the essential elements of the code.

- **Semantic Analysis:**

Semantic analysis is the second phase of the compiler and involves checking the meaning and logic of the code. This phase ensures that the code adheres to the semantic rules of the Java language. It involves checking for proper variable declarations, type compatibility, and adherence to the language's syntactic and semantic rules. Semantic analysis identifies and reports errors related to the misuse of variables, incompatible data types, undeclared identifiers, etc.

- **Memory Analyzer:**

The memory analyzer is a specialized component that focuses on memory-related aspects of the code. It checks for memory allocation and deallocation issues, such as memory leaks or attempts to access unallocated memory. This phase may involve tracking variables, ensuring they are properly initialized and released, and managing dynamic memory allocation if applicable. The memory analyzer contributes to the overall reliability and efficiency of the generated code.

- **Simplified flow of your mini compiler:**

Input: Java source code.

Scanner: Tokenizes the code and removes comments and whitespace.

Semantic Analysis: Checks the code for adherence to language rules, proper variable usage, and type compatibility.

Memory Analyzer: Focuses on memory-related aspects, checking for allocation and deallocation issues.

Output: Reports any errors or warnings found during the scanning, semantic analysis, and memory analysis phases.