Homework 1: Lexical Analyzer [*]
CS 421 Compiler Design and Construction

Asad Tariq & Fahad Shaikh

Habib University

Fall 2022

Due: September 11, 2022

With reference to the **Lexical Analyzer** implemented for the language `TUPLE`, let us, *the programmers*, elucidate some of the assumptions taken, conventions followed and references used. They are as follows:

- Some modifications were made to the given lexical specification - specified on ***pg.2*** of the assignment write-up. These changes are with respect to the **Keyword** and **Data Types** categories - upon examination of the test files, it was seen that the common language keywords and data types, `print`, `return` and `str` were not specified within the lexical specification thus, they were catered to as such per our own prerogative.
- With reference to entries within the symbol table, the decision was made not to include numeric constants such as: `2.3`, `-5`, `10` etc. Per the current implementation of lexical analyzer, the symbol table only records identifiers.
- As of now, all arithmetic operations are dealt with as part of a single category. Delineation of arithmetic operators with respect to precedence[1] is not implemented.
- For purposes of better readability and ease in visual-inspection-based verification of the output of the lexical analyzer, the generated token stream is not written to the relevant `.out` file in a single line (i.e., a literal single stream). Rather, the generated `*.out` files have the token stream printed out line-by-line. This is purely done to make verification of the generated stream easier.
- This reference proved useful when starting out with implementation of the analyzer.

[*] This document is with reference to Part 2 of the assignment.

[1] As discussed in the lecture session of 9^{th} September.