Compiler Construction

Assignment # 01 – Lexical Scanner

By Mr. Usman Wajid

1. To build the Lexical Scanner for own programming language, define the following (Regular Expression or descriptive definition),
    a. Rules for identifier name
    b. Reserve words including data types such as int, float, string, if, else etc
    c. Operators
    d. Parathesis
    e. Symbol used for end statement (use any symbol other than ; (semi-colon))

2. Draw a single DFA for your own language (use JFLAP). See "CC lecture 08" (page 3) for the DFA designed for the simple scanner

3. Write a lexical analyzer of Scanner code for a DFA designed in step 2. That is capable of the following,
    a. The scanner should take the source code from a text file and store it in an input tape, such as character array.
    b. The input tape (char array) should have two pointers, i.e, start_point and end_point, you can declare both of them as integer variables. The start_point will keep track of starting symbol index from the input tape. Similarly, the end_point will keep track of the ending symbol index of the character that has been read so far from the input tape.
    c. Exclude spaces, tabs or line breaks.
    d. Assign tokens for each valid lexeme (words)
    e. Store these tokens in a symbol table that could be any database, simple excel-sheet or even a text file

4. WHAT SHOULD YOU SUBMIT?
    a. JFLAP DFA design (Screen shot only)
    b. Your Scanner code file (Simple C language code is recommended, such as scanner.c)
    c. Sample input file (that contains some dummy code for scanning).
    d. Sample output (screen shot only)

NOTE: *There is **zero tolerance** policy on **plagiarized assignments**.*
*Also assume that your system may crash, internet may down or there will electricity shutdown in the last days of submission. So, submit it as soon as possible to ignore any inconvenience. **Late submissions** will result in reduction **-1 mark per day**.*