**15CS 385 COMPILER DESIGN LAB**

**LAB SHEET 2**

1. Associated with every programming language compiler there is a program named *recognizer* that takes a string, say string S, as input and answers "YES" if S is a sentence of the language and “NO” otherwise. Implement a Java Program to identify the following
2. Identifier
3. Function Name
4. if keyword
5. Single line and multiline Comments
6. Write a lexical analyser in Java. The analyser should read the input string (program or program fragments) from a file and determine the lexemes and their corresponding token classes/types.

For example,

if the input is:

if (txn >= 20)

rxn = txn \* 100;

else rxn = txn - 10;

the output should be saved in a file as:

<IF,if> <LPAREN,(><ID,txn> <GEQ,>=><NUM,20><RPAREN,)><ID,rxn>

<ASSIGN,=><ID,txn><TIMES,\*><NUM,100><SEMICOLON,;><ELSE,else>

<ID,rxn><ASSIGN,=> <ID,txn><MINUS,-><NUM,10><SEMICOLON,;>

**Explanation:**

1. For every lexeme in the input string, the output should be < tokentype, lexeme >.
2. White spaces - space, tabs or new lines - should be ignored. The output should be one continuous line.
3. Implement a method getToken() which returns the next token. This method can be called till EOF is reached.
4. Note that you need to look ahead one character to determine the end of a lexeme.
5. Implement a hash table with key=keyword/operator and value=tokentype. Once you

determine the lexeme, you can lookup and return the token type, if it exists.

1. If a lexeme cannot be determined to be of any token type or an illegal character is encountered, the program should report an error ”Invalid token” and exit.
2. Use the following conventions for token type:

