**COMPILER DESIGN LAB 1**

**03/02/2022**

1. Design a lexical analyzer that could ignore redundant spaces, tabs, new lines and comments in a source program (C language).
2. Read an input C file. Design a lexical analyzer that could recognize keywords, identifiers and numeric data which is valid in C language. You may restrict the length of identifiers to some reasonable value (like 32). Display appropriate message if identifiers are not valid or it is too lengthy. List out the token names along with the recognized lexemes. Construct a symbol table which holds information (name, datatype, offset, size, scope) on valid identifiers.
3. Design a lexical analyzer that could recognize the operators in C language. Display the name of the operation along with the recognized operator symbol.
4. Write a Lex program that accepts all strings of a's and b's that do not contain the subsequence abb.
5. Write a Lex program that copies a C program, replacing each instance of the keyword float by double.

\*\*\*\*\*\*\*\*\*\*