## Tutorial 1 – Lexical Analysis

1. A lexical analyser reads the source code line by line.

a) True

b) False

1. Finite State Machines can have an unlimited number of states. ( ) TRUE () FALSE
2. A regular expression is a type of pattern used to classify lexemes. () TRUE ( ) FALSE
3. You can change state in a DFA without reading any input character. ( ) TRUE () FALSE
4. In a compiler the module that checks every character of the source text is called\_\_\_\_\_\_\_\_
5. The task of the lexical analysis phase is
   1. To parse the source program into tokens of the language
   2. To build a literal table and identifier table
   3. To build a symbol table
   4. All of these
6. The number of tokens in the following C statement is

printf ( "i = %d, &i = %x", i , & i ) ;

1. Another Name for Lexical Analyser\_\_\_\_\_\_\_\_\_\_
2. Which of the following characters are ignored while lexical analysis?
   1. .
   2. =
   3. #
   4. WhiteSpace
3. Create regular expressions that specify the following (simplified) Prolog language token types
   1. **Integers** are sequences of digits[0-9] , possibly preceded by the minus sign (-).

**23, -98,0,**

* 1. **Floating-point numbers** are sequences of digits containing a decimal point, possibly preceded by the minus sign.
  2. **Octal numbers** consist of a prefix 0o (Zero-o), and a sequence of octal digits (from 0 to 7), the whole possibly preceded by the minus sign.
  3. **Atoms** are character sequences
     1. beginning with a lowercase letter and consisting only of letters, digits and underscores, or
     2. zero or more letters, digits and underscores enclosed in single quotes.
  4. **Variables** are also sequences of letters, digits and underscores that begin with an uppercase letter or with an underscore to distinguish them from atoms

You may use the Kleene Star (**\***), alternation (**|**), concatenation, and grouping with parentheses. ( )

e.g., you may use ‘[X0-9x-z]’ to represent ‘(X|0|1|2|3|4|5|6|7|8|9|x|y|z)’. Do not use any other constructs.

1. Divide the following C++ program:

float limitedSquare(x) float x; {

/\* returns x-squared, but never more than 100 \*/

return (x<=-10.0||x>=10.0)?100:x\*x;

}

into appropriate lexemes. Output the Token **<*token\_name, lexeme>***of the above program?