Name: _____

Quiz Sample Questions

COMP.3010 – Organization of Programming Languages

Spring 2019 – Dr. Wilkes

1. (2 pts) (**multiple choice**) Which of the following is **not** necessary for specification of a regular expression used to generate tokens of a programming language?
   - *y* a. Alternation
   - *y* b. Concatenation
   - *n* c. Recursion
   - *y* d. Repetition
   - e. None of the above
2. (2 pts) (**multiple choice**) (**CIRCLE ALL THAT APPLY**) A DO loop header in Fortran (before Fortran 90) requires the scanner to look-ahead multiple characters because:
   - a. Variables need not be declared
   - b. Spaces may occur within identifiers
   - c. The terminator of the DO loop is simply a label, which the parser can ignore
   - d. None of the above

T 3. (1 pt) (**true/false**) An LL parser recognizes a program via a top-down, predictive technique.

T 4. (1 pt) (**true/false**) Every LR grammar has a corresponding LL grammar.

5. (4 pts) (**short answer**) Write a regular expression – **and/or** draw a finite state automaton – that generates C-strings.
   Recall that C-strings are delimited by double quotes ("); may not contain newline characters; and may contain double-quote or backslash characters if and only if those characters are “escaped” by a preceding backslash.
   [**Hint:** Use the expression **not**(x_1, x_2, …, x_n) as shorthand for (y_1 | y_2 | … | y_m), where the y_j values are all of the characters in the character set *other than* the x_i values. For example, **not**(*, /) means “any character except * or /”. Also, use **nl** to represent the newline character.]

“ (not(nl, ” ,\)|\ “ |\\)* “