

# CS323 2025F Quiz #1

There are ten questions in this quiz.

1. Please put down your student ID. \*

2. What is the primary output of a compiler's lexical analyzer? \*

- ☐ Parse Tree
- ☐ Abstract Syntax Tree
- ☒ Sequence of Tokens
- ☐ Optimized Intermediate Code

3. Which of the following is a key responsibility of the symbol table management? \*

- ☐ Generating target machine code
- ☐ Allocating memory for variables at runtime
- ☒ Storing and retrieving information about identifiers
- ☐ Removing white space and comments

4. A finite automaton is most directly used in which phase of compilation? \*

- ☐ Syntax Analysis
- ☐ Semantic Analysis
- ☐ Code Optimization
- ☒ Lexical Analysis

5. Which concept is used to define the structure of programming language constructs like expressions and statements? \*

- ☐ Regular Expressions
- ☐ Finite Automata
- ☒ Context-Free Grammars
- ☐ Directed Acyclic Graphs

6. The string "123.45e-6" would most likely be recognized by a lexical analyzer as what? \*

- ☐ An Integer
- ☒ A Floating-Point Number
- ☐ An Identifier
- ☐ A Keyword

7. In a context-free grammar, a production that results in the same non-terminal on the left-hand side as the first symbol on the right-hand side (e.g.,  $A \rightarrow Aa$ ) is said to be: \*

- ☐ Right-recursive
- ☐ Ambiguous
- ☒ Left-recursive
- ☐ Unreachable

8. Which parsing technique builds the parse tree from the leaves (tokens) to the root (start symbol)? \*

- ☐ Top-down Parsing
- ☐ Backtracking Parsing
- ☒ Bottom-up Parsing
- ☐ Recursive Descent Parsing

9.A grammar that produces two or more distinct parse trees for the same input string is called: \*

- ☐ Recursive
- ☐ Left-factored
- ☐ Ambiguous
- ☐ Non-deterministic

10.Which of the following is a common tool used to automatically generate lexical analyzers from a specification of tokens? \*

- ☐ YACC
- ☐ Bison
- ☐ Lex
- ☐ LLVM

11.In a predictive LL(1) parser, the parse table is constructed using which function to determine which production to apply? \*

- ☐ FOLLOW
- ☐ FIRST and FOLLOW
- ☐ GOTO
- ☐ ACTION