

1.	These are nearly always based on a formal description of the syntax of programs, usually in form of a context-free grammar or BNF.	Syntax analyzers
2.	The _____ analyzer deals with small-scale language constructs, such as names and numeric literals. The _____ analyzer deals with large-scale constructs, such as expressions, statements, and program units.	Lexical, syntax
3.	Three reasons why lexical analysis is separated from syntax analysis	Simplicity, efficiency, portability
4.	One of the earliest uses of pattern matching was with text editors such as the _____	ed line editor
5.	A lexical analyser serves as the _____ of a syntax analyzer	Front end
6.	A lexical analyzer collects input characters into groups (_____) and assigns an internal code (a _____) to each group.	Lexeme, token
7.	The lexical analyzer inserts lexemes for user-defined names into the _____, which is used in the later phases of the compiler.	Symbol table
8.	The lexical-analysis process includes skipping _____ as they are not relevant to the meaning of the program.	Comments and white spaces
9.	Approaches to building a lexical analyzer: Write a formal description of the _____ of the language and use a software tool to automatically generate a lexical analyzer. The oldest tool available to generate such is called _____. Design a _____ that describes the token patterns of the language and write a program that implements the diagram, and/or hand-construct a table-driven implementation of the diagram.	Token patterns Lex State transition diagram
10.	A state transition diagram, or state diagram, is a directed graph. The nodes are labeled with _____. The arcs are labeled with _____. An arc may also include _____ when the transition is taken.	State names Input characters Actions to be done
11.	The lexeme being built by the lexical analyzer, which could be implemented as a _____, will be named _____.	Character string/array, lexeme
12.	This utility subprogram used by the lexical analyzer computes the token code for single-character tokens (parentheses and arithmetic operators).	lookup()
13.	Though names and reserved words have similar patterns, is it efficient to build a state transition diagram that checks whether the string pertains to a reserved word? What could be a possible alternative?	No – use a lookup in a table of reserved words instead
14.	The part of the process of analyzing syntax is referred to as syntax analysis, a.k.a. _____.	Parsing
15.	Parsers for programming languages construct _____ for given programs.	Parse tree
16.	_____ parsers build the tree from the root downward to the leaves. _____ parsers build the tree from the leaves upward to the root.	Top-down Bottom-up
17.	A top-down parser builds a parse tree in a _____ traversal; its actions correspond to a _____ derivation.	Preorder, leftmost
18.	A _____ is coded directly from the BNF description of the syntax of a language.	Recursive-descent parser
19.	Both the coded and tabular version of #18 are called LL algorithms. The two Ls stand for:	Left-to-right scan of input Leftmost derivation
20.	The parse order of a bottom-up parser corresponds to the reverse of a _____ derivation.	Rightmost
21.	A given right sentential form may include more than one RHS from the grammar. The correct RHS to reduce is called the _____.	Handle
22.	The worst-case complexity of common parsing algorithms is _____, making them impractical for compiler use.	$O(n^3)$
23.	Parsing algorithms used in commercial compilers have complexity _____.	$O(n)$

24.	For each terminal symbol in the RHS, that terminal symbol is compared with _____. If they do not match, it is a _____. If they match, the lexical analyzer is called to get to the next input token	nextToken, syntax error
25.	For each nonterminal, the _____ for that nonterminal is called.	Parsing subprogram
26.	The left recursion in the rule $A \rightarrow A + B$ is called _____, because it occurs in one rule.	Direct left recursion
27.		
28.		
29.		
30.		
31.	_____ are abstractions of von Neumann architecture – _____: stores both instructions and data – _____: provides operations for modifying the contents of memory	Imperative languages Memory Processor
32.	The abstractions in a language for the memory cells of a machines are _____	Variables
33.	Variables are characterized by a collection of properties or _____ – The most important of which is _____, a fundamental concept in programming languages, which we must consider scope, lifetime, type checking, initialization, and type compatibility	Attributes type
34.	A string of characters used to identify some entity in a program	name
35.	A _____ is a special word of a programming language that cannot be used as a name.	Reserved word
36.	The move from machine language to _____ was largely one of replacing absolute numeric memory addresses for data with names, making programs more readable and easier to write and maintain	Assembly language
37.	A variable can be characterized as a sextuple of attributes which are	Name, address, value, type, lifetime, scope
38.	The _____ of a variable is the machine memory address with which it is associated	Address
39.	The address of a variable is sometimes called its _____, because the address is what is required when the name of a variable appears in the left side of an assignment	l-value
40.	When more than one variable name can be used to access the same location, the variables are called _____	aliases
41.	The type of a variable determines the _____ of values the variable can store as well as the _____ that are defined for values of the type.	Range, set of operations
42.	It is the contents of the memory cell or cells associated with the variable.	Value
43.	A _____ is an association, such as between an attribute and an entity, or between an operation and a symbol.	Binding
44.	A binding is _____ if it first occurs before run time and remains unchanged throughout program execution.	Static
45.	• An _____ declaration is a program statement used for declaring the types of variables. • An _____ declaration is a default mechanism for specifying types of variables (the first appearance of the variable in the program. Both of these declarations create _____ bindings to types.	Explicit Implicit Static
46.	How are implicit declarations detrimental to reliability?	Cannot check typo and programmer errors
47.	Implicit variable type binding is done by the _____.	Language processor
48.	In _____, the context is the type of the value assigned to the variable assigned to the variable in a declaration statement (Like var in C#).	Type inference
49.	(T or F) The r-value of a variable can be accessed without determining the l-value	F

50.	With dynamic type binding, the type of a variable is not specified by a declaration statement, nor can it be determined by the spelling of its name. Instead, the variable is bound to a type when it is assigned a value in an ____.	Assignment statement
51.	Name a programming language using static type binding and another for dynamic	Static – C, VB, Dynamic – Python, JS
52.	Name a disadvantage to dynamic type binding.	reliability – error-detection capability for static bindings lost, cost
53.	It is the time during which the variable is bound to a specific memory location.	Lifetime
54.	Static variables are bound to memory cells ____ execution begins and remains bound to the same memory cell throughout execution.	before
55.	(T or F) There is no run-time overhead is incurred for allocation and deallocation of static variables.	T
56.	(T or F) Storage cannot be shared among variables of static types	T
57.	These are variables whose storage bindings are created for variables when their declaration statements are elaborated, but whose types are statically bound.	Stack-dynamic variables
58.	Stack-dynamic variables are allocated from the ____	Run-time stack
59.	The variable declarations that appear at the beginning of a Java method are ____ when the method is ____ and the variables defined by those declarations are ____ when the method completes its execution.	Elaborated, Invoked, deallocated
60.	These are nameless memory cells that are allocated and deallocated by explicit run-time instructions, specified by the programmer, which take effect during execution. These vars, which are allocated from and deallocated to the heap, can only be referenced through ____ or ____.	Explicit heap-dynamic variables Pointers Reference variables
61.	These variables are bound to heap storage only when they are assigned value. Allocation and deallocation are caused by assignment statements. All their attributes are ____ every time they are assigned.	Implicit heap-dynamic variables, bound
62.	The scope of a variable is the ____ in which the variable is visible. A variable is ____ in a statement if it can be referenced in that statement.	Range of statements visible
63.		
64.		
65.		