1. Which of the following concept of FSA is used in the compiler?
a) Code optimization
b) Code generation
c) Lexical analysis
d) Parser
Answer: c
Explanation: Because the lexer performs its analysis by going from one stage to another.

2. Which of the following is a part of a compiler that takes as input a stream of characters and produces as output a stream of words along with their associated syntactic categories?
a) Optimizer
b) Scanner
c) Parser
d) None of the mentioned
Answer: b
Explanation: A compiler's scanner scans a character-based input stream and creates a word-based output stream, with each word identified with its Syntactic category.

3. Let L1 = {w ∈ {0,1}∗ | w has at least as many occurrences of (110)'s as (011)'s}.
Let L2 = { ∈ {0,1}∗ | w has at least as many occurrences of (000)'s as (111)'s}.
Which of the following is correct?
a) L2 is regular
b) L1 and L2 are regular
c) L1 is regular but not L2
d) None of them are regular
Answer: c
Explanation: Let's look at the string 011011011011 to see if L1 is regular. The number of times 011 has occurred is four, however, it has also occurred three times. We can also produce a 110 if the string ends with 011. The following string is 110110110110, where 110 appears four times and 011 appears three times, already satisfying the.

4. What is CFG?
a) Regular Expression
b) Compiler
c) Language expression
d) All of the mentioned
Answer: b
Explanation: They're defined by the rule A->b, where A isn't terminal and b is.

5. Which of the following is a correct statement?
I. For some programming languages, there are parsing algorithms with an O(3) complexity.
II. A recursive programming language can be constructed with static storage allocation.
III. In the context of bottom-up parsing, no L-attributed definition can be evaluated.
IV. Code-improvement modifications can be carried out at both the intermediate and source code levels.
a) I and III
b) I and IV
c) I, II and IV
d) I, II, III and IV
6. Which of the following is correct regarding an optimizer Compiler?
a) Optimize the code
b) Is optimized to occupy less space
c) Both of the mentioned
d) None of the mentioned
Answer: d
Explanation: An optimising compiler is a computer programme that strives to minimise or maximise specific characteristics of an executable programme.
7. Which of the following error can Compiler diagnose?
a) Logical errors only
b) Grammatical and logical errors
c) Grammatical errors only
d) All of the mentioned
Answer: c
Explanation: Only syntactical errors can be detected by the compiler.
8. In which of the following phase of the compiler is Lexical Analyser?
a) Second
b) Third
c) First
d) All of the mentioned
Answer: c
Explanation: Lexical Analyzer is the First Phase of the Compiler.
9. Which of the following does an address code involve?
a) No unary operators
b) Exactly 3 address
c) At most Three address
d) None of the mentioned
Answer: d
Explanation: In computer science, three-address is an intermediate code used by optimizing compilers to aid in the implementation of code-improving transformations.

10, An object module for a group of programs that were compiled separately is handed to a linker. Which of the following about an object module isn't true?
a) Relocation bits
b) Names and locations of all external symbols denied in the object module
c) Absolute addresses of internal symbols
d) Object code
Answer: c
Explanation: A linker, sometimes known as a link editor, is a computer program that merges one or more object files generated by a compiler into a single executable, library, or another object file.
11. Characters are grouped into tokens in which of the following phase of the compiler design?
a) Code generator
b) Lexical analyzer
c) Parser
d) Code optimization
Answer: b
Explanation: Gives tokens as output
12. Why Generation of intermediate code based on an abstract machine model is useful in compilers?
a) Writing for intermediate code generation
b) Portability of the front end of the compiler
c) Implementation of lexical analysis and syntax analysis is made easier
d) All of the mentioned
Answer: c
Explanation: Intermediate code generator receives input from its predecessor phase, semantic analyzer, in the form of an annotated syntax tree.
13. Why System program such as compiler are designed?
a) They are Serially usable
b) They are Re-enterable
c) They are Nonreusable
d) All of the mentioned
Answer: b
Explanation: Re-enterable is the keyword for compiler being designed.
14. Which of the following technique is used for building cross compilers for other machines?
a) Canadian Cross
b) Mexican Cross
c) X-cross
d) Brazilian Cross
Answer: a
Explanation: The Canadian Cross is a technique for building cross compilers for other machines. Given three machines X, Y, and Z, one uses machine X (e.g. running Windows XP on an IA-32 processor) to build a cross compiler that runs on machine Y (e.g. running Mac OS X on an x86-64 processor) to create executables for machine

15.Which of the following can detect an error if a programmer by mistake writes multiplication instead of division?
a) Interpreter
b) Compiler or interpreter test
c) Compiler
d) None of the mentioned
Answer: d
Explanation: No Logical errors can't be detected.
16. The grammar S → aSa | bS | c is?
a) LL(1) but not LR(1)
b) LR(1) but not LR(1)
c) Both LL(1) but not LR(1) & LR(1) but not LR(1)
d) None of the mentioned
Answer: c
Explanation:
First(aSa) = a
First(bS) = b
First(c) = c
LR parsers are more powerful than LL (1) parsers and LR (1).

17. Recursive descent parsing is an example of ____________
a) Top down parsing
b) Bottom up parsing
c) Predictive parsing
d) None of the mentioned
Answer: a
Explanation: Top down is the answer.
18. LR stands for ___________
a) Left to right
b) Left to right reduction
c) Right to left
d) Right most derivation and Left to right and a in reverse
Answer: d
Explanation: Right most derivation and left to right and in reverse is used for LR.
19. Which is the most powerful parser?
a) SLR
b) LALR
c) Canonical LR
d) Operator-precedence
Answer: c
Explanation: Canonical tops all other parsers.
20. When will the relationship between '+' and '-' be <?
a) For unary minus
b) Minus is right associative

c) All of the mentioned
d) None of the mentioned
Answer: c
Explanation: Both statements are true.