Reg. No.															

Question Paper Code	13276

B.E. / B.Tech. - DEGREE EXAMINATIONS, NOV / DEC 2024

Fifth Semester

Computer Science and Business Systems

20CBPC501 - COMPILER DESIGN

Regulations - 2020

Duration: 3 Hours Max. Marks: 100

PART - A (MCQ) (20 × 1 = 20 Marks)

Answer ALL Questions

		Marks	*K – Level*	*CO*
1.	Which computer program accepts the high-level language and converts it into assembly language? (a) Interpreter (b) Linker (c) Assembler (d) Compiler	*1*	*K1*	*CO1*
2.	Which errors can be checked by a Compiler? (a) Logical Error (b) Syntax Error (c) Run time error (d) Both (a) and (b)	*1*	*K1*	*CO1*
3.	Which phase of the compiler groups the characters into tokens? (a) Scanner (b) Parser (c) Code Optimizer (d) Code Generator	*1*	*K1*	*CO1*
4.	The transitional function of a NFA is (a) Q X Σ→Q (b) Q X Σ→2^Q (c) Q X Σ→2^n (d) Q X Σ→Qn	*1*	*K2*	*CO1*
5.	Bottom up parser generates _____ (a) Leftmost derivation (b) Rightmost derivation (c) Leftmost derivation in reverse (d) Rightmost derivation in reverse	*1*	*K1*	*CO2*
6.	Which is the most powerful parser? (a) SLR parser (b) CLR parser (c) LALR parser (d) LL parser	*1*	*K1*	*CO2*
7.	A top-down parser generates (a) Left-most derivation in reverse (b) Left-most derivation (c) Right-most derivation in reverse (d) Right –most derivation	*1*	*K1*	*CO2*
8.	Which grammar gives multiple parse trees for the same string? (a) Unambiguous (b) Regular (c) Ambiguous (d) All of the above	*1*	*K1*	*CO2*
9.	An attributed grammar is a (a) A grammar that defines rules for error handling (b) A formal way to define semantics by associating attributes with grammar productions (c) A technique for code optimization (d) A method used to generate machine code	*1*	*K2*	*CO3*
10.	In syntax-directed definitions, what are the two types of attributes used? (a) Inherited and Logical (b) Semantic and Syntactic (c) Synthesized and Inherited (d) Static and Dynamic	*1*	*K1*	*CO3*
11.	Which of the following component is important for semantic analysis? (a) YACC (b) LEX (c) Symbol Table (d) Type Checking	*1*	*K1*	*CO3*
12.	What is the primary purpose of a symbol table in a compiler? (a) To optimize the code (b) To store information about variables, functions, and objects (c) To generate intermediate code (d) To provide an execution environment for the program	*1*	*K1*	*CO3*
13.	Which of the following is an intermediate representation used in compilers? (a) Abstract Syntax Tree (b) Source code (c) Machine code (d) Object code	*1*	*K1*	*CO4*
14.	In control-flow analysis, which structure is used to represent the flow of a program? (a) Directed Acyclic Graph (b) Control Flow Graph (c) Syntax Tree (d) Finite State Machine	*1*	*K1*	*CO4*

15. Which of the following is an example of global optimization? *1 K1 CO4*
(a) Loop invariant code motion (b) Constant Propagation
(c) Common sub expression elimination (d) Strength reduction

16. Peep-hole optimization is a technique used in compilers to *1 K1 CO4*
(a) Remove redundant code at the global level
(b) Improve code within a small window or sequence of instructions
(c) Eliminate unused variables
(d) Reorganize loops for better performance

17. What is the primary goal of instruction scheduling in pipeline architectures? *1 K1 CO5*
(a) To minimize memory usage
(b) To reduce execution time by avoiding pipeline stalls
(c) To increase the number of instructions in a program
(d) To improve the readability of the code

18. Which of the following is a key technique used for register allocation? *1 K1 CO5*
(a) Graph coloring (b) Loop unrolling (c) Peephole optimization (d) Strength reduction

19. Which of these features is commonly associated with object-oriented programming languages? *1 K1 CO5*
(a) Direct memory access (b) Data abstraction and encapsulation
(c) Low-level instruction manipulation (d) Recursive function optimization

20. How does compilation of non-imperative programming languages differ from imperative languages? *1 K2 CO5*
(a) It eliminates the need for control flow structures
(b) It focuses more on mathematical functions and expressions rather than state changes
(c) It generates machine code without an intermediate representation
(d) It uses more registers for storing variables

PART - B (10 × 2 = 20 Marks)

Answer ALL Questions

21. What is a lexeme? *2 K1 CO1*
22. List out the three parts of lex program. *2 K1 CO1*
23. Show whether the given grammar is ambiguous or not. *2 K2 CO2*
S ->aSb | ε
24. Write the rule for left recursion and illustrate it with an example. *2 K2 CO2*
25. Differentiate synthesized and inherited attributes with example. *2 K1 CO3*
26. Define Annotated Parse Tree. *2 K1 CO3*
27. What is dead code elimination? *2 K2 CO4*
28. Define three address codes. *2 K1 CO4*
29. Compare Non Imperative with Imperative programming languages. *2 K2 CO5*
30. List out the types of loop optimizations for cache memory. *2 K1 CO5*

PART - C (6 × 10 = 60 Marks)

Answer ALL Questions

31. a) Explain the various phases of compiler with neat diagram and translate the following statement c:=a-b*21. *10 K2 CO1*

OR

b) Show the minimal DFA for the given regular expression (a*/b*)*a. *10 K2 CO1*

32. a) Construct SLR (1) parser for the following grammar: *10 K3 CO2*
S->0A1 | 0B1 | 1A | 1B2

OR

b) Construct a predictive parser for the following grammar: *10 K3 CO2*
E→E + T | T
T→T * F | F
F→ (id)
Also parse the input string id+id*id.

33.	a)	Explain Syntax Direct Definition with examples.	*10*	*K2*	*CO3*
		OR			
	b)	Summarize the parameter passing in Run Time Environment.	*10*	*K2*	*CO3*
34.	a)	Apply the various ways of representing intermediate code with neat examples.	*10*	*K3*	*CO4*
		OR			
	b)	Discuss in detail about global data flow analysis.	*10*	*K3*	*CO4*
35.	a)	Explain in detail about (i) Type Systems (ii) Data Abstraction	*10*	*K2*	*CO5*
		OR			
	b)	Explain a simple code generator algorithm with suitable example.	*10*	*K2*	*CO5*
36.	a) i)	Explain Loop unrolling and Loop fusion in detail.	*5*	*K2*	*CO4*
	ii)	Discuss any three issues in the design of a code generator.	*5*	*K2*	*CO5*
		OR			
	b) i)	Explain common subexpression elimination with example.	*5*	*K2*	*CO4*
	ii)	Summarize the compilation of object oriented features.	*5*	*K2*	*CO5*