Reg. No.																

Question Paper Code	14279

B.E. / B.Tech. - DEGREE EXAMINATIONS, NOV / DEC 2025

Sixth Semester

Computer Science and Engineering

(Common to Computer Science and Engineering (IoT))

20CSPC602 - COMPILER DESIGN

Regulations - 2020

Duration: 3 Hours Max. Marks: 100

PART - A (MCQ) (10 × 1 = 10 Marks)

Answer ALL Questions

Marks K – Level CO

1. Which computer program accepts the high-level language and converts it into assembly language? *1 K1 CO1*
   (a) Interpreter (b) Linker (c) Assembler (d) Compiler
2. Which of the following is the correct order of the phases in a traditional compiler? *1 K1 CO1*
   (a) Lexical analysis, Semantic analysis, Code optimization, Code generation
   (b) Lexical analysis, Syntax analysis, Semantic analysis, Code generation
   (c) Syntax analysis, Lexical analysis, Code optimization, Code generation
   (d) Code generation, Code optimization, Lexical analysis, Syntax analysis
3. Which of the following is NOT a responsibility of the lexical analyzer? *1 K1 CO2*
   (a) Tokenizing the input (b) Removing comments and whitespace
   (c) Performing syntax analysis (d) Identifying keywords and operators
4. In the context of lexical analysis, which of the following is an example of a token? *1 K1 CO2*
   (a) Keyword if (b) Integer value 42 (c) Identifier x (d) All of the above
5. What does the syntax analyzer detect in a program? *1 K1 CO3*
   (a) undefined variables (b) type mismatches (c) syntax errors (d) run-time errors
6. Which of the following best describes a predictive parser? *1 K1 CO3*
   (a) It is a type of bottom-up parser.
   (b) It uses the first symbol of the input to decide which rule to apply.
   (c) It can handle any context-free grammar.
   (d) It constructs the entire parse tree bottom-up.
7. Which of the following is an example of an intermediate representation (IR) in a compiler? *1 K1 CO4*
   (a) Assembly language (b) Abstract Syntax Tree (AST)
   (c) Bytecode (d) Machine code
8. Why is intermediate code used in the compilation process? *1 K1 CO4*
   (a) To enable the compiler to run faster
   (b) To perform machine-independent optimizations
   (c) To avoid errors during machine code generation
   (d) To minimize the number of symbols in the source code
9. Which of the following is the main responsibility of a code generator in a compiler? *1 K1 CO5*
   (a) Detecting semantic errors in the program
   (b) Translating the intermediate code into a target machine code or assembly code
   (c) Performing high-level optimizations
   (d) Generating the abstract syntax tree (AST)
10. Which optimization technique improves the performance of a program by reducing the number of instructions executed in a loop? *1 K1 CO6*
    (a) Inlining (b) Loop unrolling (c) Register allocation (d) Constant folding

PART - B (12 × 2 = 24 Marks)

Answer ALL Questions

11. Define Symbol Table. *2 K1 CO1*
12. Define Grouping of the phases of the compiler. *2 K2 CO1*

13. Compare the features of NFA and DFA. *2 K2 CO2*

14. Define tokens, patterns and lexemes. *2 K1 CO2*

15. Demonstrate the rules to construct FIRST Function and FOLLOW Function. *2 K2 CO3*

16. Convert Non Deterministic Grammar to Deterministic Grammar. *2 K3 CO3*
S->iEtS | iEtSeS | a
E->b

17. Outline some examples of static checking. *2 K2 CO4*

18. Show the three address codes for the following. *2 K2 CO4*
x=a+(b*-c)+(d*-e). Represent the three address codes by Triples.

19. List the issues in the design of code generators. *2 K1 CO5*

20. Outline the DAG for the statement a = (a*b+c)–(a*b+c). *2 K2 CO5*

21. Apply the basic block concepts, how would you represent the dummy blocks with no statements indicated in global dataflow analysis? *2 K3 CO6*

22. List out the properties of optimizing compilers. *2 K2 CO6*

PART - C (6 × 11 = 66 Marks)

Answer ALL Questions

23. a) Explain the phases of a compiler indicating the inputs and outputs of each phase in translating the statement "amount=principle+rate*36.0". *11 K2 CO1*

OR

b) (i) Explain the language processing system with a neat diagram. *6 K2 CO1*

(ii) Explain various Errors encountered in different phases of compiler. *5 K2 CO1*

24. a) (i) Illustrate in detail about the role of Lexical analyzer with the possible error recovery schemes. *6 K2 CO2*

(ii) Outline in detail about issues in lexical analysis. *5 K2 CO2*

OR

b) Demonstrate DFA construction using Direct method for the following Regular Expression (a|b)*abb. *11 K2 CO2*

25. a) Construct Stack implementation of shift reduce parsing for the grammar *11 K3 CO3*
E->E+E
E->E*E
E-> (E)
E->id and the input string id1+id2*id3.

OR

b) Construct the Predictive Parser for the following grammar. *11 K3 CO3*
S→ (L) | a
L→ L, S | S
Parse the input and give a rightmost derivation for (a, (a, a)) and show the handle of each right-sentential form.

26. a) Make use of backpatching to generate code for Boolean expressions and flow of control statements. *11 K3 CO4*

OR

b) Construct intermediate code for the following code segment *11 K3 CO4*
if (a > b) x = a + b else x = a – b.

27. a) Infer activation record and Explain stack allocation of activation records using examples. *11 K2 CO5*

OR

b) Compare and contrast of static, stack and Heap allocation. *11 K2 CO5*

28. a) Explain global data flow analysis with necessary equations. *11 K2 CO6*

OR

b) Explain the principal Sources of Optimization strategies with appropriate examples. *11 K2 CO6*