

C

Reg. No. :

--	--	--	--	--	--	--	--	--	--	--	--	--

Question Paper Code: U6201

B.E. / B.Tech. DEGREE EXAMINATION, APRIL/MAY 2025

Sixth Semester

Computer Science And Engineering

21UCS601- PRINCIPLES OF COMPILER DESIGN

(Regulations 2021)

Duration: Three hours

Maximum: 100 Marks

Answer ALL Questions

PART A - (5 x 1 = 5Marks)

1. If the lexical analyzer finds a lexeme with the same name as that of a reserved word. What Analyzer will do? CO1-U
(a) overwrites the word (b) overwrites the functionality
(c) Overwrites the error (d) None of the above
2. The basic actions in the parsing process of any bottom-up parser are CO1-U
(a) Shift, Produce, accept, reject (b) Shift, Produce, reject, accept
(c) Rotate, produce, accept, reject (d) Shift, reduce, accept, reject
3. In programming, Boolean expressions are used ----- CO1-U
i. To Compute the logical values
ii. As conditional expressions in flow control statements
(a) i only (b) ii only (c) i and ii (d) none.
4. In activation record, Which of the following Stores the address of activation record of the caller procedure? CO1-U
(a) Access Link (b) Actual Parameters (c) Control Link (d) Temporaries
5. The graph that shows basic blocks and their successor relationship is called----. CO1-U
(a) DAG (b) Flow graph (c) control graph (d) Hamiltonion graph

PART – B (5 x 3= 15 Marks)

- | | | |
|-----|--|-----------|
| 6. | Define tokens, patterns and lexemes with suitable example. | CO1-U |
| 7. | Draw the syntax tree of the statement $a^* - (b + c)$ | CO 2- App |
| 8. | What are the intermediate languages. | CO1-U |
| 9. | Distinguish between static vs dynamic storage allocation. | CO1-U |
| 10. | What is peephole? what is the need of peephole optimization? | CO1-U |

PART – C (5 x 16= 80 Marks)

- | | | | |
|-----|---|---------|------|
| 11. | (a) Explain the various phases of a compiler in detail. Also write down the output for the following expression: position: =initial + rate * 60. | CO2-App | (16) |
| | Or | | |
| | (b) Derive DFA for the regular expression $a^*b(a+b)^*$ | CO2-App | (16) |
| 12. | (a) Design non-recursive predictive parser for the following grammar and parse the string $id+id*id$.
$E \rightarrow E+T \mid T$ $T \rightarrow T * F \mid F$ $F \rightarrow (E) \mid id$ | CO2-App | (16) |
| | Or | | |
| | (b) Find the SLR parsing table for the given grammar
$E \rightarrow E+E \mid E * E \mid (E) \mid id$ and parse $(a+b*c)$ | CO2-App | (16) |
| 13. | (a) Explain in detail the various representation of intermediate code. | CO1-U | (16) |
| | Or | | |
| | (b) Explain the translation of Boolean expressions in detail. | CO1-U | (16) |
| 14. | (a) Illustrate in detail about the code generation algorithm with an example. | CO2-App | (16) |
| | Or | | |
| | (b) Develop a quicksort algorithm for reads nine integers into an array and sorts them by using the concepts of activation tree. | CO2-App | (16) |
| 15. | (a) Explain the principal sources of optimization in detail. | CO1-U | (16) |
| | Or | | |
| | (b) Discuss the three different storage allocation techniques. | CO1-U | (16) |