

Reg. No. :

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

Question Paper Code: U6201

B.E./B.Tech. DEGREE EXAMINATION, NOV 2025

Sixth Semester

Computer Science Engineering

21UCS601- PRINCIPLES OF COMPILER DESIGN

(Regulation 2021)

Duration: Three hours

Maximum: 100 Marks

PART A - (5 x 1 = 5 Marks)

1. How many tokens the statement $sum = sum + I;$ has? CO1- U
(a) 5 (b) 6 (c) 10 (d) 15
2. Which of the following are kernel items? CO1- U
i) $E \rightarrow E + T$ ii) $S' \rightarrow S$ iii) $T \rightarrow T * F$ iv) $.(E)$
(a) i,ii only (b) i,iii only (c) ii and iii only (d) i and iv only
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. Which of the following known as the text part of a program that does not change at runtime. Its memory requirements are known at the compile time? CO1- U
(a) Code (b) Procedures (c) Variables (d) All of the above
5. What is the equivalent optimized expression of $x = x * 2$? CO1- U
(a) $x = x / 2$ (b) $x = x * 0.5$ (c) $x = x + x$ (d) $x = x - x$

PART – B (5 x 3= 15 Marks)

6. Explain the role of lexical analyzer. CO1 -U
7. What is Handle Pruning? CO2 -U
8. What is synthesized and inherited attributes? CO1 -U
9. What are the limitations of stack allocation? CO1-U
10. What is peephole and what is the need of peephole optimization? CO1-U

PART – C (5 x 16= 80 Marks)

11. (a) Construct an NFA equivalent to the regular expression $(0+1)^*(00+11)(0+1)^*$ CO2-App (16)
- Or
- (b) Write the regular expression for the pattern starting with “#” and ending with “;” with middle portion is the pattern for identifier. Derive the DFA for the pattern. CO2-App (16)
12. (a) Check whether the following grammar is a LL (1) grammar. CO2-App (16)
 $S \rightarrow iEts \mid iEtSeS \mid a$
 $E \rightarrow b$
and Parse the string “ibtibtaea”
- Or
- (b) Construct LALR parsing table for the grammar CO2-App (16)
 $S \rightarrow CC$
 $C \rightarrow cC \mid d$
13. (a) Explain about Type checking and Type Conversion with examples CO1-U (16)
- Or
- (b) Explain the translation of Boolean expressions in detail. CO1-U (16)
14. (a) Describe the issues in design of code generator. CO1-U (16)
- Or
- (b) Explain in detail about storage allocation strategies. CO1-U (16)
15. (a) What happens if you do not optimize the code? If you optimize what are the issues associated with the optimization? What are the principle sources of optimization? Explain with example. CO1- U (16)
- Or
- (b) Explain the principal sources of optimization in detail. CO1- U (16)