PVP 20

Code: 20CS3601

III B.Tech - II Semester – Regular Examinations – JUNE 2023

COMPILER DESIGN
(COMPUTER SCIENCE & ENGINEERING)

Duration: 3 hours Max. Marks: 70

Note: 1. This paper contains questions from 5 units of Syllabus. Each unit carries 14 marks and have an internal choice of Questions.
2. All parts of Question must be answered in one place.

BL – Blooms Level CO – Course Outcome

			BL	CO	Max. Marks
		UNIT-I			
1	a)	Discuss the phases of a compiler indicating the inputs and outputs of each phase in translating the statement "**A = P + R * 45** ".	L2	CO1	10 M
	b)	Describe the role of lexical analysis in compiler design.	L2	CO1	4 M
		OR			
2	a)	Discuss about the input buffering scheme in lexical analyzer.	L2	CO1	10 M
	b)	Describe specification and recognition of tokens.	L2	CO1	4 M
		UNIT-II			
3	a)	Calculate FIRST and FOLLOW for the following grammar: E-> E+T/T T-> T*F/F F-> (E)/id	L3	CO2	7 M

	b)	Explain the error recovery in predictive parsing.	L3	CO2	7 M
		OR			
4	a)	i. Differentiate Top Down Parser And Bottom Up Parser? Give example for each. ii. Describe a context free grammar?	L2	CO2	4 M 3 M
	b)	i. Sketch syntax tree for the expression $a=b*-c+b*-c$. ii. Construct the algorithm for FIRST and FOLLOW in parser.	L3	CO2	4 M 3 M
		UNIT-III			
5	a)	Consider the following grammar: E-> E+E E-> E*E E->id Construct shift reduce parsing of the input string *"$id_1+id_2+id_3$"*.	L3	CO3	7 M
	b)	i. Explain why SLR and LALR are more economical to construct than canonical LR(CLR)? ii. Explain what is meant by **goto** function in LR parser? Give an example	L4	CO5	4 M 3 M
		OR			
6	a)	Consider the following grammar. $S \rightarrow AS / b$, $A \rightarrow SA / a$ Construct the ***SLR*** parse table for the	L3	CO3	10 M

		grammar. Show the actions of the parser for the input string ***abab.***			
	b)	(i) Compare the types of LR parsers. (ii) Explain what is LR(k) parsing?	L4	CO5	2 M 2 M
		UNIT-IV			
7	a)	Construct CLR Parsing table for the given grammar: S → CC , C → aC / d	L3	CO3	7 M
	b)	Construct Three Address Code for the following expression: *(a * b) + (c + d) – (a + b + c + d)*	L3	CO4	7 M
		OR			
8	a)	Show the following grammar is **LALR(1)** S → Aa / bAc / dc / bda A → d	L3	CO3	7 M
	b)	Explain the different storage allocation strategies.	L3	CO4	7 M
		UNIT-V			
9	a)	What is DAG and flow graph? Explain their role in compilation process.	L3	CO4	7 M
	b)	Explain the main issues in code generation.	L3	CO4	7 M
		OR			
10	a)	Explain various machine **independent code** optimization techniques.	L3	CO4	7 M
	b)	Explain various machine **dependent code** optimization techniques.	L3	CO4	7 M