CS61

USN	1	M	S							

RAMAIAH Institute of Technology

(Autonomous Institute, Affiliated to VTU)
(Approved by AICTE, New Delhi & Govt. of Karnataka)
Accredited by NBA & NAAC with 'A+' Grade

SUPPLEMENTARY SEMESTER EXAMINATIONS – JULY 2023

Program	:	**B.E. – Computer Science and Engineering**	**Semester**	:	**VI**
Course Name	:	**Compiler Design**	**Max. Marks**	:	**100**
Course Code	:	**CS61**	**Duration**	:	**3 Hrs**

Instructions to the Candidates:

- Answer one full question from each unit.

UNIT - I

1.	a)	Examine the translation operations of code optimizer and code generator with appropriate examples.	CO1	(08)
	b)	Relate the usages of tokens, patterns with lexemes of lexical analyzer operations.	CO1	(05)
	c)	Appraise about the operations of phases of compiler with respect to the translation of following statement. K=(k+5)/(g-h).	CO1	(07)
2.	a)	Discuss recognition of reserved words and identifiers with examples.	CO1	(08)
	b)	Demonstrate the ambiguous grammar and its drawbacks with an example. Write the rules for the "dangling – else" grammar and discuss. How it can be rewritten as a an unambiguous grammar.	CO1	(07)
	c)	Illustrate the error detection and recovery strategies of parser with examples.	CO1	(05)

UNIT - II

3.	a)	What is meant by handle pruning? Show the working of a shift reduce parser for accepting $id_1 * id_2$, considering the grammar: $E \rightarrow E + T \mid T$ $T \rightarrow T * F \mid F$ $F \rightarrow id$	CO2	(06)
	b)	Given the grammar: $S \rightarrow a \mid (L)$ $L \rightarrow L, S \mid S$ i) Do the necessary changes to make it suitable for LL(1) parser. ii) Construct the predictive parsing table. Check the resultant grammar is LL(1) or not.	CO2	(10)
	c)	With an example explain the working of Recursive Descent parsing.	CO2	(04)
4.	a)	Construct LR (1) parsing table for the grammar. $S \rightarrow Aa \mid bAc \mid Bc \mid bBa$ $A \rightarrow d$ $B \rightarrow d$ Show the pasring steps for the input string "bda".	CO2	(12)
	b)	Explain the rules for computing FIRST and FOLLOW. Illustrate using the grammar $S \rightarrow AB$ $A \rightarrow Ca \mid \varepsilon$ $B \rightarrow BaAC \mid c$ $C \rightarrow b \mid \varepsilon$	CO2	(08)

UNIT - III

5.	a)	Distinguish between synthesized and inherited attributes. Consider the grammar below for unsigned binary numbers. Design an SDD for the same to computenum. vaI, the decimal value of an input string. Draw a dependency graph based on the SDD for the inputstring 1101. *num→numdigit	digit* *digit →0|1*	CO3	(10)
	b)	Write the recursive code fragment for finding factorial of a number. Construct Activation Tree and Activation record for the same.	CO3	(10)

6.	a)	Design a Syntax Directed Definition for structure of an array type. Illustrate its applications on input " int a[10] [10]" D→T / C T→ int / float C→ [num]C C→ε	CO3	(08)
	b)	Illustrate how the desk calculator is implemented on a bottom-up parsing stack with the help of semantic actions.	CO3	(06)
	c)	Generate the SDT for the following grammar **S→while (C) S;**	CO3	(06)

UNIT- IV

7.	a)	Construct syntax directed definition for the following flow control statement. S →if (B) S_1 else S_2 S → while (B) S_1	CO4	(10)
	b)	Name the different type of representation of 3-address code and translate the given arithmetic expression into each type. a – b * c + d – a + b	CO4	(10)
8.	a)	What is DAG? Write DAG for the expression. ((x + y) – ((x + y) * (x – y))) + ((x+ y) * (x – y))	CO4	(06)
	b)	Write and explain the syntax directed definition for switch statement.	CO4	(08)
	c)	Write the algorithm for unification of a pair of nodes in a type graph.	CO4	(06)

UNIT - V

9.	a)	Discuss in detail the issues in the design of a code generator.	CO5	(10)
	b)	Construct the DAG for the basic block: d = b * c e = a + b b = b * c a = e - d Simplify the above TAC assuming (a} only a is live on exit from the block (b} a, b, and c are live on exit from the block.	CO5	(05)
	c)	List the rules to be followed when reconstructing the basic blocks from aDAG.	CO5	(05)

10.	a)	Code Fragment :
		{ prod=1;
		i =0;
		while(i<=10){
		prod=prod*a[i];
		i++;
		}
		i. Obtain the three address code for the code
		ii. Construct the Basic Blocks and flow Graph for the code
		iii. Identify the loops in the flow graph.
			CO5	(10)

	b)	Illustrate and generate the machine code for a=b+c by explaining the functions of address and register descriptor.	CO5	(10)

****