Roll No.

ANNA UNIVERSITY (UNIVERSITY DEPARTMENTS)

B.E - (Full Time) - END SEMESTER EXAMINATIONS, NOV/DEC 2024

COMPUTER SCIENCE AND ENGINEERING
Semester - V
CS6109 - Complier Design
(Regulation RUSA 2018)

Time: 3hrs **Max. Marks: 100**

CO 1	Comprehensively identify the issues in every phase of the compiler
CO 2	Analyse the design issues in the different phases of the compiler and design the phases by integrating appropriate tools
CO 3	Articulate the front end in the design of a compiler with, Lexical, Syntactic and Semantic phases and its processes for any given source language
CO 4	Analyse and understand the various code optimizations that are necessary for any given intermediate code or assembly level code for sequential algorithms
CO 5	Apply and design code optimization techniques for any input code with error recovery
CO 6	Design a compiler by incorporating the various phases of the compiler for any new source language

BL – Bloom's Taxonomy Levels
(L1 - Remembering, L2 - Understanding, L3 - Applying, L4 - Analyzing, L5 - Evaluating, L6 - Creating)

PART- A (10 x 2 = 20 Marks)
(Answer all Questions)

Q. No	Questions	Marks	CO	BL
1	Write code using lex to eliminate blank space in a given string	2	CO1	L2
2	What is a Synchronization token in error recovery?	2	CO1	L1
3	State the defects when a grammar is ambiguous? How to avoid ambiguity.	2	CO2	L2
4	Is LALR (1) $\subseteq$ LR (1)? Yes/No, give explanation?	2	CO2	L1
5	What is a display? Write with example its access control method.	2	CO3	L2
6	What is the risk involved in runtime management of a compiler?	2	CO3	L2
7	What is the Marker terminal symbol? What is its purpose?	2	CO4	L2
8	Why is DAG an important data structure in the design of a compiler? Which phases are dependent on this structure?	2	CO5	L2
9	Give any two machine idioms.	2	CO4	L1
10	What converts the Java bytecode to target machine instructions?	2	CO6	L2

PART- B (8 x 8 = 64 Marks)
(Answer any 8 questions)

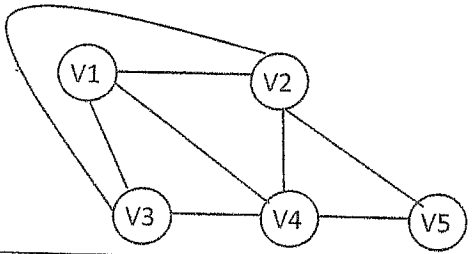
Q. No	Questions	Marks	CO	BL
11.	Write in detail on the Front-end and Back-end of a compiler.	8	CO1	L3
12.	Find the DFA for the RE (b|a)|(a*)(ab)*#	8	CO2	L3

13.	For the ambiguous grammar:
	`S → if E then S | if E then S else S | a`
	`E → b`
	i. Show how an LR(1) parser resolves the ambiguity between "dangling else" constructs.
	ii. Construct the LR(1) parsing table and identify the relevant lookaheads
		4+4	CO3	L5

Q. No	Questions	Marks	CO	BL
14.	i. Write algorithm to construct FIRST and FOLLOW sets for a CFG and Find the same for the CFG given below: ii. Construct LL parser for the given grammar. S → A C | B D A → a | ε B → b | ε C → c D → d	4+4	CO3	L5
15.	The CFG to translate hexadecimal to decimal is given below, apply syntax directed definition (SDD) and annotate the grammar. Show how the conversion of hex code: 1A3 to decimal takes place using the generated SDD? Give details of the attributes (values and functions) and its types as applied for this conversion. H → H D H → D D → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F	8	CO3	L4

16.	Translate the following C code to three-address statement using backpatching.
	Write the semantics and draw the parse tree
	if ((i != j) && (i<n) && (!m))
	{ x = n; n = j; p = ++i; }
	else
	{ x = m; m = i; p = --j; }
		8	CO4	L4

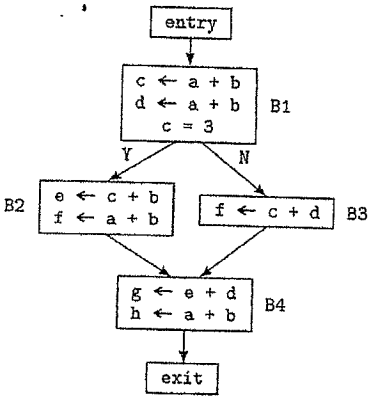
17.	Explain simple code generation from DAG and detail the address and register descriptors by applying it to the basic block given below: (Assume 2 registers are provided). t1 = a + b t2 = a + b t3 = t1 + t2	8	CO4	L5
18.	A real array A[i, j, k] has index i ranging from 1 to 4, index j ranging from 0 to 4, and index k ranging from 5 to 10. Reals take 8 bytes each. Suppose array A is stored starting at byte 0. i) Find the location of: **A[3][4][5] = A[4][3][9] + A[1][2][7]** ii) Generate 3 address code for statement(i) & draw the parse tree.	8	CO4	L5
19.	Assuming that function widen in a language such as C can handle any of the types in the type hierarchy, translate the expression: x = (s + c) * (t + d) Assume that c and d are characters, s and t are short integers, and x is a float. Write the widening conversion function and write the semantics for type conversion in translation.	8	CO6	L4
20.	Develop an SDT scheme to generate code using the backpatching technique for a repeat – until loop. See that your solution works for the case of nested loops and *break* and *continue* statements at different nesting levels also. **(1) S → repeat M_1 L until M_2 E;** **(2) S → continue;** **(3) S → break;** **(4) L_1 → S ; M L_2** **(5) L → S** **(6) M → ε**	8	CO5	L5

Q. No	Questions	Marks	CO	BL
21.	Generate optimal code using Dynamic Programming technique for the statement: **x = (a * b)/((c * d) * e) * f.** Assume 3 registers are available and instructions and their associated costs as given below: **Instruction** — **Cost** **Ri = Ri op Rj** — **1** **Ri = Ri op Mj** — **2** **Ri = Rj** — **1** **Ri = Mj** — **1** **Mi = Rj** — **1**	8	CO5	L5
22.	Write the graph coloring algorithm and apply it to the Register interference Graph (RIG) given below, assume 3 registers are provided for allocation. 	8	CO5	L5

PART- C (2 x 8 = 16 Marks)

Q. No	Questions	Marks	CO	BL

23.	Given the following C code, show how all the phases of a compiler performs? Write the outcome of each phase lexical analysis to code generation (Assume target machine with two registers R1 and R2)
	`1. #include <stdio.h>`
	`2. int main() {`
	`3. int a=12, b = 10, c = 0;`
	`4. a = a + b;`
	`5. c = a + b + a + b;`
	`6. C = a + b;`
	`7. return 0;`
	`}`
		8	CO6	L6

24.	Apply the any 2 Optimizations to the following flow graph and explain any one in detail. Write the algorithm and show final optimized flow graph. 	8	CO6	L5