RAGHU ENGINEERING COLLEGE (AUTONOMOUS)
(Approved by AICTE, New Delhi, Accredited by NBA (CIV, ECE, MECH & CSE), NAAC with 'A+' grade
& Permanently Affiliated to JNTU-Gurajada, Vizianagaram)
Dakamarri, Bheemunipatnam Mandal, Visakhapatnam Dist. – 531 162 (A.P.)
Ph: +91-8922-248001, 248002 Fax: + 91-8922-248011
E-mail: principal@raghuenggcollege.com website: www.raghuenggcollege.com

BTECH II YEAR I SEMESTER (AR 20)

Name of the Subject : **Compiler Design**
Subject Code : 20CS4009
Name of the Subject Coordinator : **Mrs.M.Dhanalakshmi**

QUESTION BANK

S.No	QUESTIONS	Level	Course Outcome	MARKS
Language Processors: Introduction, the structure of a compiler, the science of building a compiler, programming language basics. Lexical Analysis: The role of the lexical analyzer, input buffering, recognition of Tokens, the lexical analyzer generator lex program specification, finite automata, from regular expressions to automata, design of a lexical-analyzer generator, optimization of DFA-based pattern matchers.				
1	What is a Language Processing System? And explain the language processing system in detail.	L1	CO1	4M
2	Explain the following: Lexeme, Token and pattern?	L1	CO1	4M
3	Write Transition diagram to recognize: a) pipe, pet, item, temper, perpetual b) all, mall, fatal, llama, lame	L1	CO1	4M
4	Explain different compiler construction tools	L1	CO1	4M
5	Explain application of Compiler Technology	L2	CO1	7M
6	What is the role of a lexical analyzer? And also explain about Lexical Analyzer functionalities, Lexical errors?	L2	CO1	7M
7	Explain about the 'lex' tool structure with an example program?	L2	CO1	7M
8	Explain about symbol table management and error detection and handling?	L2	CO1	7M
9	Define a compiler. Explain the structure of a compiler with an example using neat sketch?	L4	CO1	10M
10	What is a Translator? Explain the difference between Compiler and Interpreter.	L2	CO1	10M
11	Explain how input buffering helps to speed up the reading of source program?	L4	CO1	10M
12	Derive the regular expression for the tokens given below and also draw a transition diagram to recognize the following tokens: i) Relational operators ii) Integer constant iii) Identifier iv) White spaces v) Exponent part of a number	L4	CO1	10M
Unit-2				
Syntax Analysis: Introduction, context-free grammars (CFG), derivation, top-down parsing, recursive and non recursive top down parsers, bottom-up parsing, Operator precedence parser, Introduction to LR parsing: simple LR parser, more powerful LR parsers, using ambiguous grammars, parser hierarchy, and automatic parser generator YACC tool.				
1	Explain the procedure for elimination of common factors (left factoring and left recursion with an example.	L1	CO2	4M
2	What is operator Precedence Parser Explain with Example?	L1	CO2	4M

3	Find the FIRST and FOLLOW set for the below grammar G. E → TE' E' → +TE' | ε T → FT' T' → *FT' | ε F → (E) | id	L1	CO2	4M
4	Compute FIRST and FOLLOW for each of the non-terminals: E→TE' E'→+E | ε T→FT' T'→T | ε F→PF' F'→*F' | ε P →(E) | a | b | c	L1	CO2	4M
5	What is Recursive Descent Parser? Explain with an example.	L2	CO2	7M
6	Construct predictive parsing table for dangling-else problem check if the grammar is LL(1) G: S → iEtSS' | a S' → eS | ε E → b	L2	CO2	7M
7	Show error recovery moves on an erroneous input by a predictive parser	L2	CO2	7M
8	Consider the grammar: S → (L) | a L → L,S | S i. Make necessary changes to make it suitable for LL(1) ii. Check the resultant grammar is LL(1) or not iii. Show the actions of the LL(1) parser, given the input is ((a),a)	L2	CO2	7M
9	Consider following Grammar: G: E → E+T|T T → T*F|F F → (E)|id i. Let us construct LR(0) items and Parsing Table ii. Draw DFA for the set of items iii. Parse the input string id * id + id	L4	CO2	10M
10	Construct LALR(1) parsing table for the following grammar E →(E) | id	L4	CO2	10M
11	Consider the grammar S → aAd | bBd | aBe | bAe A → c B → c i. Check if it is an CLR(1) grammar ii. Construct LALR(1) Parsing table			
12	Construct LR(1) Parsing table for the Following Grammar? S→CC C→cC C→d	L4	CO2	10M
Unit-3				

Syntax-Directed Definitions: Introduction, evaluation orders for SDD's, applications of syntax-directed translation, syntax-directed translation schemes, and implementing Lattributed SDD's. Intermediate-Code Generation: variants of syntax trees, three-address code, types and declarations, type checking, control flow statements, switch-statement, and procedures.

1	Construct the Abstract Syntax Tree and DAG for the below Grammar and for the following expression a = (-b) * d + c + (-b) * d **Production** — **Semantic Rules** 1) E -> E1+T — E.node= make Node('+', E1.node,T.node) 2) E -> E1-T — E.node= makeNode('-', E1.node,T.node) 3) E -> T — E.node = T.node 4) T -> (E) — T.node = E.node 5) T -> id — T.node = make Leaf(id, id.entry) 6) T -> num — T.node = make Leaf(num, num.val)	L1	CO3	4M
2	What are the properties of three address code format? And construct the quadruples three address code format for the below expression: A = B * (-C) + B * (-C)	L1	CO3	4M
3	Compare all the three address code formats?	L1	CO3	4M
4	What are the properties of three address code format? And construct the quadruples, , Indirect Triples, three address code format for the below expression: A = B * (-C) + B * (-C)	L2	CO3	7M
5	Explain the differences between Parse Tree and Syntax Tree with a suitable example.	L2	CO3	7M
6	Explain conversion of Control Flow Statements to Three Address Code?	L2	CO3	7M
7	Explain about the S-attributed, L-attributed definitions grammar with an example.	L3	CO3	10M
8	Explain about Synthesized ,Inherited attributes with an example.	L3	CO3	10M
9	Explain the differences between SDD and SDT.	L2	CO3	10M

Unit-4

Run-Time Environments: Storage organization, stack allocation of space, access to nonlocal data on the stack, heap management, introduction to garbage collection, introduction to tracebased collection. Machine Independent Code optimizations: The principal sources of optimization, introduction to data-flow analysis, foundations of data-flow analysis, constant propagation, partialredundancy elimination, and loop optimization in flow graphs.

1	Explain differences between Static scoping and Dynamic scoping?	L1	CO4	4M
2	Explain about the Position of machine independent code optimizer?	L1	CO4	4M
3	What is the run-time environment? And discuss its requirement?	L1	CO4	4M
4	Explain about the static space allocation in runtime storage, in-detail?	L2	CO4	7M
5	Discuss about the activation record?	L2	CO4	7M
6	How non-local data is accessing between procedures? Explain with an example.	L2	CO4	7M
7	How the activation tree is useful for stack memory allocation method? Explain with suitable example.	L4	CO4	10M
8	What are the issues considered at the time of code optimization? Explain them?	L3	CO4	10M

9	What is constant propagation, partial redundancy elimination? Explain with an example?	L4	CO4	10M

Unit-5

Code Generation: Issues in the design of a code generator, the target language, addresses in the target code, basic blocks and flow graphs, optimization of basic blocks, a simple code generator. Machine Dependent Code Optimizations: peephole optimization, register allocation and assignment, code generation algorithm.

1	What are the issues considered at the time of code optimization? Explain them.	L1	CO5	4M
2	Explain about the Position of code generator?	L1	CO5	4M
3	Explain about the Position of machine dependent code optimizer?	L1	CO5	4M
4	Explain about the register allocation and assignment with an example?	L4	CO5	7M
5	Discuss the difference between machine dependent and machine independent code optimization?	L1	CO5	7M
6	Write a Three address code for the following control statements i) Simple if ii) If else iii) While	L2	CO5	7M
7	Explain about basic blocks, Flow Graphs with an example.	L3	CO5	10M
8	Which issues are considered at the time of code generation? Explain them.	L1	CO5	10M

9	Apply the Structure-Preserving Transformations (or) Peephole optimization (or) Principle sources of optimization for the following C code.
	for i from 1 to 10 **do**
	for j from 1 to 10 **do**
	$a[i, j] = 0.0;$
	for i from 1 to 10 **do**
	$a[i, i] = 1.0;$
		L3	CO5	10M

Level :L1:Remembering ; **L2**: Understanding; **L3**: Applying; **L4**: Analyzing; **L5**: Evaluating
(Refer REVISED Bloom's Taxonomy Action Verbs)

UNIT WISE QUESTIONS IN THE QUESTION BANK MUST COVER ENTIRE SYLLABUS