Course Code: 20CS0516 **R20**

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY:: PUTTUR
Siddharth Nagar, Narayanavanam Road – 517583

QUESTION BANK (DESCRIPTIVE)

Subject with Code : COMPILER DESIGN (20CS0516)

Course & Branch: B.Tech – CSE,CSIT **Year & Sem:** II/I
Regulation: R20

UNIT –I
INTRODUCTION AND LEXICAL ANALYSIS

1	**a**	What do you understand by language processor?	[L2][CO1]	**[2M]**
	b	Describe about different language processors used in compiler design	[L2][CO1]	**[4M]**
	c	Give the differences between compiler and interpreter.	[L4][CO1]	**[6M]**
2	**a**	Define compiler.	[L1][CO1]	**[2M]**
	b	Analyse the process of compilation while designing a compiler.	[L4][CO2]	**[10M]**
3	**a**	List all the phases of compiler	[L1][CO2]	**[2M]**
	b	Give the neat diagram of phase of a compiler	[L2][CO2]	**[4M]**
	c	Explain each phase of a compiler.	[L2][CO2]	**[6M]**
4		Design the compiler by using the source program position=intial+rate*60.	[L6][CO3]	**[12M]**
5	**a**	Analyze the reasons for separating the lexical analysis and syntax analysis.	[L4][CO2]	**[4M]**
	b	Illustrate the steps involved in designing the compiler by using the source program a=b+c*10.	[L3][CO3]	**[8M]**
6	**a**	Describe Bootstrapping	[L2][CO1]	**[8M]**
	b	Explain the different applications of compiler technology	[L2][CO1]	**[4M]**
7	**a**	Discuss the Compiler construction Tools	[L2][CO3]	**[6M]**
	a	Differentiate tokens, patterns, and lexeme.	[L4][CO1]	**[6M]**
8	**a**	Explain in detail about the role of lexical analyzer in Compiler Design.	[L2][CO1]	**[6M]**
	b	Write about input buffering?	[L3][CO1]	**[6M]**
9		Discriminate the following terms a) Specification of Tokens b) Recognition of Tokens	[L5][CO1]	**[12M]**

10	**a**	What is LEX	[L2][CO3]	**[2M]**
	b	Explain the working of a LEX Tool	[L2][CO3]	**[6M]**
	c	Give the structure of LEX program	[L2][CO3]	**[4M]**

Course Code: 20CS0516 **R20**

UNIT –II
SYNTAX ANALYSIS AND TOP DOWN PARSING

1	**a**	Explain the role of parser.	[L2][CO1]	**[4M]**
	b	Define Context Free Grammar with example.	[L1][CO1]	**[4M]**
	c	Compare left most and right most derivations with examples	[L4][CO1]	**[4M]**
2	**a**	Define parse tree.	[L1][CO2]	**[2M]**
	b	Construct Leftmost and Rightmost derivation and parse tree for the string 3*2+5 from the given grammar. Also check it's ambiguity for Set of alphabets $\sum$ = {0,…,9, +, *, (,)} E → I E →E + E E →E * E E →(E) I →ε | 0 | 1 | … | 9	[L6][CO2]	**[10M]**
3	**a**	Define Ambiguity.	[L1][CO1]	**[2M]**
	b	Interpret how to eliminate ambiguity for the given Ambiguous Grammar.	[L3][CO1]	**[10M]**
4	**a**	What is left recursion? Describe the procedure of eliminating Left recursion.	[L5][CO1]	**[4M]**
	b	Eliminate left recursion for the following grammar E→E+T/T T→T*F/F F→(E)/id	[L1][CO1]	**[4M]**
	c	Show what you understand by Left factoring. Perform left factor for the grammar A→abB/aB/cdg/cdeB/cdfB	[L2][CO1]	**[4M]**
5	**a**	List the types of Parsers available	[L1][CO2]	**[4M]**
	b	Design the recursive decent parser for the following grammar E→ E+T/T T→T*F/F F→(E)/id	[L6][CO3]	**[8M]**
6	**a**	What is meant by Non-recursive predictive parsing	[L2][CO3]	**[2M]**
	b	Illustrate the rules to be followed in finding the FIRST and FOLLOW.	[L3][CO1]	**[6M]**
	c	Find FIRST and FOLLOW for the following grammar? E→ E+T/T T→T*F/F F→(E)/id	[L3][CO2]	**[4M]**

7		Consider the grammar S→AB | ABad A→d E →b D→b | ε B→c Construct the predictive parse table and check whether the given grammar is LL(1) or not.	[L6][CO3]	**[12M]**
8		Consider the grammar E→TE' E'→+TE' | -TE' | ε T→FT' T'→*FT' | / FT' | ε F→GG' G'→^F / ε G→(E) / id Calculate FIRST and FOLLOW for the above grammar and Construct LL(1) Table for the above grammar.	[L4][CO2]	**[12M]**
9		Consider the grammar E→E+T/T, T→T*F/F, F→(E)|id Design predictive parsing table and check the given grammar is LL(1) or not?	[L6][CO3]	**[12M]**
10	**a**	Discuss the types of errors.	[L2][CO2]	**[6M]**
	b	Explain Error recovery in predictive parsing with an Example.	[L2][CO2]	**[6M]**

Course Code: 20CS0516 **R20**

UNIT –III
BOTTOM UP PARSING AND SEMANTIC ANALYSIS

1	**a**	Explain about handle pruning	[L2][CO1]	**[6M]**
	b	Summarize about LR parsing	[L2][CO1]	**[6M]**
2	**a**	Describe bottom up parsing	[L1][CO2]	**[4M]**
	b	Differences between SLR, CLR, LALR parsers	[L4][CO2]	**[8M]**
3		Prepare Shift Reduce Parsing for the input string using the grammar S→(L)|a L→L,S|S a. (a,(a,a)) b. (a,a)	[L6][CO3]	**[12M]**
4	**a**	Define augmented grammar.	[L1][CO2]	**[2M]**
	b	Construct the LR(0) items for the following Grammar S→L=R / R L→*R / id R→L	[L6][CO3]	**[10M]**
5		Construct SLR Parser for the following grammar E→E+T / T T→TF / F F→F* / a / b	[L6][CO3]	**[12M]**
6		Construct CLR Parsing table for the given grammar S→CC C→aC/d	[L6][CO3]	**[12M]**
7		Design the LALR parser for the following Grammar S → AA A → aA A →b	[L6][CO3]	**[12M]**
8	**a**	What is YACC parser?	[L1][CO3]	**[2M]**
	b	Explain in detail the processing procedure of YACC Parser generator tool.	[L2][CO3]	**[6M]**
	c	How YACC will resolve the parsing action conflicts and the error recovery.	[L2][CO3]	**[4M]**
9	**a**	Explain syntax directed definition with example	[L2][CO2]	**[6M]**
	b	Define a syntax-directed translation and explain with example.	[L2][CO2]	**[6M]**
10	**a**	Give the evaluation order of SDD with an example.	[L5][CO2]	**[6M]**
	b	Discuss Type Checking with suitable examples.	[L2][CO4]	**[6M]**

Course Code: 20CS0516 **R20**

UNIT –IV

INTERMEDIATE CODE GENERATION AND RUN TIME ENVIRONMENT

1	**a**	What do you understand by Intermediate Code	[L2][CO5]	**[2M]**
	b	Analyse different types of Intermediate Code with an example.	[L4][CO5]	**[10M]**
2	**a**	List and define various representation of Three Address Codes	[L1][CO5]	**[4M]**
	b	Explain representation of Three Address Codes with suitable Examples	[L2][CO5]	**[8M]**
3		Produce quadruple, triples and indirect triples for following expression: $(x + y) * (y + z) + (x + y + z)$	[L6][CO5]	**[12M]**
4	**a**	Describe scope and life time of variable.	[L2][CO4]	**[2M]**
	b	Illustrate Control Flow Statements.	[L3][CO4]	**[10M]**
5	**a**	Justify the need for Storage Organization.	[L6][CO4]	**[4M]**
	b	Describe the Storage Organization with simple examples.	[L2][CO4]	**[8M]**
6	**a**	List out the properties of memory management	[L1][CO4]	**[4M]**
	b	Discuss Storage allocation strategies with suitable example	[L2][CO4]	**[8M]**
7		Evaluate the following terms i. Stack allocation ii. Static allocation iii. heap allocation	[L5][CO4]	**[12M]**
8	**a**	Define Activation Record.	[L1][CO5]	**[2M]**
	b	Sketch the format of Activation Record in stack allocation and explain each field in it.	[L3][CO5]	**[10M]**
9	**a**	Discuss about symbol table entries.	[L2][CO4]	**[6M]**
	b	Describe the various operations on symbol table.	[L2][CO4]	**[6M]**
10	**a**	Define Symbol table.	[L1][CO4]	**[2M]**
	b	Explain different types of Data structure used for symbol table.	[L2][CO4]	**[10M]**

Course Code: 20CS0516 **R20**

UNIT –V

CODE OPTIMIZATION AND CODE GENERATION

1		Interpret the principles of optimization techniques to be considered during code generation.	[L3][CO5]	**[12M]**
2	**a**	Discuss about function preserving transformations.	[L2][CO6]	**[6M]**
	b	Describe about loop optimization technique.	[L2][CO5]	**[6M]**
3		Explain the following i) Basic blocks ii) Flow Graphs	[L3][CO6]	**[12M]**
4	**a**	List the optimization techniques of basic blocks	[L1][CO6]	**[4M]**
	b	Analyse different types of optimization techniques of basic blocks	[L4][CO6]	**[8M]**
5	**a**	Create the DAG for following statement. a+b*c+d+b*c	[L6][CO6]	**[6M]**
	b	Construct the DAG for the following basic blocks 1. t1:=4*i 2. t2:=a[t1] 3. t3:=4*i 4. t4:=b[t3] 5. t5:=t2*t4 6. t6:=prod+t5 7. prod:=t6 8. t7:=i+1 9. i:=t7 **if i<=20 goto 1**	[L6][CO6]	**[6M]**
6	**a**	List out the properties of global data flow analysis and explain it.	[L2][CO6]	**[6M]**
	b	Discuss about machine dependent optimization	[L2][CO5]	**[6M]**
7		Explain the peephole optimization Technique with examples.	[L2][CO5]	**[12M]**
8	**a**	List all the issues in the design of a code generator	[L2][CO6]	**[4M]**
	b	Explain the issues to be handled when code generator is designed.	[L2][CO6]	**[8M]**
9	**a**	Analyse the different forms in target program.	[L4][CO6]	**[6M]**
	b	Explain the target machine in code generator.	[L2][CO6]	**[6M]**
10	**a**	Analyze Simple code generator	[L4][CO6]	**[6M]**
	b	Evaluate Register allocation and register assignment techniques.	[L5][CO6]	**[6M]**

Prepared by: Dr. R.M Mallika, Mr.P.Krishnamoorthy, Mrs. Viswa sahithya, CSE, SIETK