IS62B3

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

EXAMINATIONS SEPTEMBER /OCTOBER 2020
SUPPLEMENTARY SEMESTER / GRADE IMPROVEMENT/ RE –REGISTERED CANDIDATES

Program	:	**B.E. : Information Science & Engineering**	**Semester**	:	**VI**
Course Name	:	**Compiler Design**	**Max. Marks**	:	**100**
Course Code	:	**IS62B3**	**Duration**	:	**3 Hrs**

Instructions to the Candidates:

- Answer any one full question from each unit.

UNIT – I

1.	a)	Explain with a neat diagram, the phases of a compiler. Mention the input and output for each phase with an example, "position = initial + rate * 60".	CO1	(10)
	b)	Differentiate between token, pattern and lexeme with examples.	CO1	(06)
	c)	Write the transition diagram for relational operators.	CO1	(04)
2.	a)	Explain the roles of lexical analyzer.	CO1	(07)
	b)	What are buffer pairs in input buffering? Explain.	CO1	(07)
	c)	Develop a Lex program to check the number is odd or even.	CO1	(06)

UNIT – II

3.	a)	Explain different error recovery strategies of compilation process.	CO2	(08)
	b)	Given the grammar: S->aABb, A->c|ε, B->d|ε i) Compute FIRST and FOLLOW sets. ii) Construct the predictive parsing table. iii) Show the moves made by the predictive parser on acdb.	CO2	(12)
4.	a)	List and explain with examples the common programming errors which can occur at different levels of compilation process.	CO2	(06)
	b)	Design an algorithm to remove left factoring from the given grammar. Apply the same to the following grammar: S -> i E t S | i E t S e S | a, E -> b	CO2	(08)
	c)	What is recursive-descent parsing? Develop a typical procedure for a non-terminal in a top down parser.	CO2	(06)

UNIT – III

5.	a)	Construct CLR parsing table for the grammar S->AA, A->aA|b. Apply CLR parsing algorithm to check the validity of the input string bab.	CO3	(12)
	b)	Explain in detail the syntax-directed definitions for while statement.	CO3	(08)

Created by trial version of DocuFreezer

6. a) Construct SLR (1) parsing table for: CO3 (10)
E → (L) | a
L → EL | E

b) Explain the purpose of using dependency graphs. CO3 (04)

c) Explain bottom-up evaluation of S-attributed definitions. CO3 (06)

UNIT – IV

7. a) Explain basic blocks and flow graphs with suitable examples. CO4 (08)

b) Translate the following expressions using translation scheme for boolean expressions: CO4 (04)
i) (a<b || a>c || a!=d)
ii) (a==b&&b==g&& j==k)

c) Explain SDT of switch statement. CO4 (08)

8. a) For the given productions shown below, write semantic rules and construct annotated parse tree for 3*5+4n CO4 (08)
L->En, E->E|+T, E->T, T->T|*F, T->F, F->(E), F->digit

b) Explain the following with examples: CO4 (06)
i) Quadruples ii) Triples

c) Discuss on backpatching in intermediate code generation. CO4 (06)

UNIT – V

9. a) Discuss the various issues in code generation. CO5 (10)

b) Formulate optimal machine code sequence for the given expression: *G=(a*a+2*a*b+b*b)* CO5 (10)

10. a) Discuss about the stack allocation with respect to code generation phase of compiler. CO5 (10)

b) Explain the general structure of activation record. Explain the purpose of each item in the activation record. CO5 (10)

****

Created by trial version of DocuFreezer