Reg. No. | | | | | | | | | | | | | |

B.E. / B.TECH. DEGREE EXAMINATIONS, MAY 2023

Sixth-Semester

CS18602 – COMPILER DESIGN

(Computer Science and Engineering)

(Regulation 2018)

TIME: 3 HOURS **MAX. MARKS: 100**

COURSE OUTCOMES	STATEMENT	RBT LEVEL
CO 1	To understand the major phases of compilation.	2
CO 2	To gain the skill to design and implement a prototype of compiler.	3
CO 3	To identify the parsers and practice the experiments.	4
CO 4	To apply the various optimization techniques.	3
CO 5	To acquire knowledge about different compiler construction tools.	3

PART- A (10 x 2 = 20 Marks)

(Answer all Questions)

		CO	RBT LEVEL
1.	What is the difference between a compiler and an interpreter give an example of a language that uses each?	1	2
2.	What do you mean by cross compiler? Give examples.	1	2
3.	List the various error recovery strategies for a lexical analysis.	2	1
4.	Write the Regular definition for the language “All strings of a’s and b’s with an even number of a’s and an odd number of b’s”.	2	3
5.	Compare SLR and CLR parsers.	3	2
6.	List down the conflicts occur in LR parsers.	3	2
7.	State the benefits of using machine-independent intermediate representations?	4	1
8.	What are the pros and cons of heap storage in memory allocation?	4	2
9.	What is data flow analysis?	5	2
10.	What is constant folding? Give example.	5	2

PART- B (5 x 14 = 70 Marks)

			Marks	CO	RBT LEVEL
11. (a)	(i)	Illustrate how the different phases of the compiler will handle the following expression: Position: =initial+ rate*60. Write down the output after each phase of compilation.	(9)	1	3

	(ii)	Illustrate diagrammatically the working of language processing system.	(5)	1	3
		(OR)			
(b)	(i)	Illustrate with an example how the different possible errors are handled in each phase of the compiler.	(10)	1	3
	(ii)	How the different phases of the compiler are grouped? Also list down the reasons for grouping the phases of a compiler.	(4)	1	3
12. (a)	(i)	Write the Regular Expression, Regular definition and LEX expression for the language defined over the alphabets Σ= {0,1} and “All strings start with 0’s and ends with 1’s.	(10)	2	2
	(ii)	Explain the need and role of a Lexical Analyzer.	(4)	2	2
		(OR)			
(b)	(i)	Write a LEX code to validate the given Arithmetic Expression.	(8)	2	2
	(ii)	Design the Lexical Analyzer for a sample Language using LEX.	(6)	2	2
13. (a)		Construct a predictive parsing table for the given grammar G, E→E+T |T, T→ T * F |F, F→ (E) |id. Also parse the string w= id*(id+id).	(14)	3	4
		(OR)			
(b)		Construct the SLR (1) parsing table for the given Grammar G, S→aB B→bB | b. Check whether the string w= abbbbb will be accepted or not.	(14)	3	4
14. (a)		For the given grammar E → E+T | E-T | T, T→ (E) | id | num. Write the Syntax Directed Definition and draw the annotated parse tree and also construct the syntax tree for an expression a+b*5.	(14)	4	4
		(OR)			
(b)		Design a type checker to handle expressions, statements and functions.	(14)	4	4

Q. Code: 210475

15. (a)	For the given code segment construct the flow graph and apply the possible optimization. SUM: =0 i: =0 do SUM: =prod + a[i]*b[i]; i: =i+1 While(i<20);	**(14)**	**5**	**3**
	(OR)			
(b)	Construct a DAG and write the optimized code for the expression x=a * (b-c) + (b-c) * d	**(14)**	**5**	**3**

PART- C (1 x 10 = 10 Marks)

(Q.No.16 is compulsory)

		Marks	CO	RBT LEVEL
16.	For the given Grammar, S→SS+ | SS* |a parse the string w=aaa*a++ using the Shift-Reduce parser to check whether the given input string is accepted or not.	**(10)**	**3**	**5**

****