VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

REG.NO.:

SLOT: A2+TA2

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - II
FALL SEMESTER 2025-2026

Programme Name & Branch : B.Tech. & CSE
Course Code and Course Name : BCSE307L and Compiler Design
Faculty Name(s) : Prof. SUGANTHINI C, Prof. NIVITHAK, Prof. VISWANATHAN A, Prof. KRISHNARAJ N, Prof. MUTHUNAGAI S U, Prof. ARUMUGA ARUN R, Prof. BHAWANA TYAGI, Prof. ISLABUDEEN M, Prof. NAGA PRIYADARSINI R, Prof. BASKARAN P, Prof. RAGHUL J, Prof. VISHNU PRIYA A,Prof. MUKHTAR AHMAD SOFI, Prof. NITHYA K, Prof. BHUVANESWARI M, Prof.KANAGARAJ.R
Class Number(s) : VL2025260101628, 1616, 1580, 1591, 1613, 1611, 1639, 1621, 1633, 1600, 1577, 1585, 6947, 1609, 158
Date of Examination : 05.10.2025
Exam Duration :90 minutes **Maximum Marks:50**

General instruction(s):

- Answer All Questions
- Provide complete solutions for all questions, including all steps
- M- Max mark; CO – Course Outcome; BL – Blooms Taxonomy Level (1 – Remember, 2 – Understand, 3 – Apply,4 – Analyse,5 – Evaluate,6 – Create)
- Course Outcomes:
  CO2: To make students familiar with lexical analysis and parsing techniques.
  CO3: To understand the various actions carried out in semantic analysis.
  CO4: To make the students get familiar with how the intermediate code is generated.
  CO5: To understand the principles of code optimization techniques and code generation.

Q. No	Question	Module	Marks	CO	BL
1.	Consider the following CFG with set of terminals **{begin, end, a}** and non terminal symbols **{Stat, Block, Body}** of the grammar and **Stat** is considered the starting symbol for this grammar. Compute the set of LR (1) items set for the following Grammar and draw the corresponding DFA diagram and parse the input string : **1. "begin begin begin a end end end"** **2. "begin a end end"** with necessary steps. **Stat -> Block** **Block -> begin Block end** **Block -> Body** **Body -> a**	2	10	2	2
2.	a) Construct an L attribute (SDD) Syntax-Directed Definition for the following Grammar G . Use Synthesized attributes to determine X.Val the decimal-number value of an input string. For example, the translation of string (Binary) 111.001 should be the decimal number 7.125. The Grammar G: (Where X is starting symbol) **X -> Y.Y / Y** **Y -> YB / B** **B -> 0 / 1**	3	5	3	3
	b) Construct an Annotated Parse tree using the above Grammar given in this question with an input (Binary) string 111.001	3	5		

VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

REG.NO.:

SLOT: A2+TA2

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - II
FALL SEMESTER 2025-2026

3.	Translate the following program code into Three-Address Code (TAC) and convert it into triples format.
	`int main() {`
	`char ch;`
	`scanf(" %c", &ch);`
	`switch (tolower(ch)) {`
	`case 'a':`
	`case 'e':`
	`case 'i':`
	`case 'o':`
	`case 'u': printf("'%c' is a vowel.\n", ch);`
	`break;`
	`default: if (isdigit(ch)) {`
	`printf("'%c' is a digit.\n", ch);`
	`} else if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z')) {`
	`printf("'%c' is a consonant.\n", ch);`
	`} else {`
	`printf("'%c' is a special character.\n",ch); }`
	`break; }`
	`return 0; }`
		4	10	4	3

4.	Consider the following C like code that involves nested for loop and array. Generate Three Address Code with back patching and Demonstrate how back patching resolves jump addresses for Nested for loop and if condition.
	`void Sort(int arr[], int n) {`
	`int i, j, temp;`
	`for (i = 0; i < n - 1; i++) {`
	`for (j = 0; j < n - i - 1; j++) {`
	`if (arr[j] > arr[j + 1]) {`
	`temp = arr[j];`
	`arr[j] = arr[j + 1];`
	`arr[j + 1] = temp;/`
	`} } }`
		4	10	4	3

5.	Consider the following C code (unoptimized). Identify the all possible principal sources of optimization in the program and write the optimized code after applying these optimizations
	`i=0;`
	`while(i<n) {`
	`j=0;`
	`while(j<m) {`
	`P[i][j] = (a + b) - (c + d);`
	`Q[i][j] = (a * b) * (a+b-8);`
	`R[i][j] = (i * 2) - (j * 2);`
	`S[i][j] = (a + b) - (c + d) + Y[i];`
	`j++; }`
	`i++; }`
		5	10	5	3

****