



VIT

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

REG.NO.: 25PCE0724

SLOT: D1

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST – I Key
WINTER SEMESTER 2025-2026

Programme Name & Branch : B.Tech & Common to all
Course Code and Course Name : BACSE104 - Structured and Object-Oriented Programming
Faculty Name(s) : Prof. SENTHIL KUMAR K / Prof. RAMESH BABU
 VEMULURI / Prof. VIJAYASHREE J / Prof. KEERTHIKA P /
 Prof. KALAIVANI K / Prof. GOUTAM MAJUMDER / Prof.
 DHIVYAA C R / Prof. SRI PREETHAA K R / Prof.
 GUNASEKAR M / Prof. SWETHA V / Prof. SYAMASUDHA
 VEERAGANDHAM / Prof. ASHOK KUMAR RAI / Prof.
 JABANJALIN HILDA J / Prof. VINILA JINNY / Prof.
 VISWANATHAN A / Prof. ARUMUGA ARUN R / Prof. VELU
 K

Class Number(s) : VL2025260501471/1480/1464/1438/5405/1478/1442
 /1461/6372/1486/1431/6755/1454/1445/1447/1458/
 1468

Date of Examination : 30/01/2026
Exam Duration : 90 minutes
Maximum Marks: 50

General instruction(s):

- Answer All Questions
- Course Outcomes:
CO1: To analyze and implement basic programming constructs, decision-making statements, and loops in C/C++
CO2: To apply modular programming techniques by utilizing functions, structures, and files for efficient problem-solving in real-world applications.

Q. No	Question	Module	Marks	CO	BL
1	Write a C program to read an integer N and then get N integer elements for an <code>input_array</code> . Using appropriate conditional statements, looping constructs, and array operations, generate an <code>output_array</code> by performing the following operations: <ul style="list-style-type: none"> • If the <code>array_element</code> is a positive number, compute the factorial of the <code>array_element</code> and store the result in the corresponding position of the <code>output_array</code>. • If <code>array_element</code> is zero, compute the sum of all elements in the <code>input_array</code> and store the result in the corresponding position of the <code>output_array</code>. • If <code>array_element</code> is a negative number, compute the factorial of the square of <code> array_element </code> and store the result in the corresponding position of the <code>output_array</code>. • Finally, display the contents of the <code>input_array</code>, and <code>output_array</code>. 	1	10	1	3
2	List the different storage classes in C. Critically analyze how the selection of storage classes affects variable lifetime, scope, initialization, memory allocation, and overall program performance. With appropriate code-level illustrations, explain how an incorrect	1	10	1	4



SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST – I Key
WINTER SEMESTER 2025-2026

u/n u
u/n u
0/0 ✓

	choice of storage class may result in logical errors, unnecessary memory consumption, or degraded execution efficiency.				
3.	a) Write a C program that defines a user-defined function to receive an integer number from the main() function and reverse the digits of the number. The user-defined function must directly modify the same on the variable used in the main() function without using any return statement. Choose the suitable function calling method to achieve this requirement. Display the value of the integer number before and after invoking the function.	1	5	1	3
	b) Write a C program to read a string from the user and perform the following operations using standard string handling functions: <ul style="list-style-type: none"> Determine and display the length of the string. Check whether the given string is a palindrome. 	1	5		
4.	a) Compare and contrast the dynamic memory allocation functions malloc() and calloc() in C with respect to their syntax, memory initialization behavior, and typical usage scenarios.	2	5		
	b) Consider the following C program. Determine the exact output produced by the program. Then, clearly explain and differentiate the behavior and functionality of the expressions *(p + 2), *(p) + 2, *p++, and *p - 2 with respect to pointer arithmetic, operator precedence, and value evaluation.			2	2
	<pre>#include <stdio.h> int main() { 0 1 2 3 4 int a[5] = {10, 20, 30, 40, 50}; int *p = a; 1 2 3 4 printf("** (p + 2) = %d\n", *(p + 2)); printf("** (p) + 2 = %d\n", *(p) + 2); printf("** p++ = %d\n", *p++); printf("** p - 2 = %d\n", *p - 2); return 0; }</pre>	2	5		
5.	Develop a C program to read a two-dimensional matrix and generate its transpose using pointers to arrays rather than direct array indexing. Display both the original matrix and the transposed matrix. Subsequently, implement matrix multiplication between the original matrix and its transpose, and display the resulting matrix. The program should clearly demonstrate pointer-based traversal of 2D arrays.	2	10	2	3

[0 =] C