REG.NO.:

SLOT: D2

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

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

Programme Name & Branch : **B.Tech. & Computer Science and Engineering**
Course Code and Course Name : **BACSE104 and Structured and Object-Oriented Programming**
Faculty Name(s) : **DHIVYAA C R, RAJARAJAN G, NARAYANAN PRASANTH**
GERALDINE BESSIE AMALI D, KALAIVANI K, SARITHA MURALI, SUNIL KUMAR, SEKARAPANDIAN N, GLADYS GNANA KIRUBA B, GUNASEKAR M, VIMALA DEVI K, NAGA RAJA G, RAMESH BABU VEMULURI, SWETHA V, GOUTAM MAJUMDER, MUCHENEDI HARI KISHOR
Class Number(s) : **VL202526050-1465,1455,1433,1444,1459,1493,1462,1451,1453, 1435,1437,1440,2341,1482,1485**
Date of Examination : **30th Jan, 2026**
Exam Duration : **90 minutes** **Maximum Marks: 50**

General instruction(s):

- Answer All Questions
- M - Max mark; CO – Course Outcome; BL – Blooms Taxonomy Level (1 – Remember, 2 – Understand, 3 – Apply, 4 – Analyse, 5 – Evaluate, 6 – Create)
- Course Outcomes (Type the CO statements covered in this question paper. Use the CO number as per the syllabus copy)
- All the programs must contain main() with out which the program is invalid

Q. No	Question	Module	Marks	CO	BL

1.	Write a C program to generate the monthly mobile bill according to the following conditions:
	• For the first 200 calls: ₹0.80 per call
	• For the next 300 calls: ₹0.50 per call
	• Above 500 calls: ₹0.30 per call
	• If the total bill exceeds ₹1000, add 10% surcharge.
	• Minimum bill: ₹150.
	Print the bill for 'n' customers in the format:
	`Enter number of customers: 2`
	`S.No Name Calls Bill`
	`---`
	`Enter customer name: Ram`
	`Enter number of calls: 176`
	`1 Ram 176 150.00`
	`Enter customer name: Shyam`
	`Enter number of calls: 874`
	`2 Shyam 874 422.20`
		1	10	1	3

2.	Explain in detail about the different storage classes available in C. Discuss how each storage class affects the scope, visibility, and lifetime of variables.	1	10	1	2
3.	Write a C program to count the number of distinct vowels (a, e, i, o, u) in a string and display each vowel with its frequency. Sample: Input → "Education is important" Output → a:2, e:1, i:2, o:1, u:1	1	10	1	3
4.	a) Distinguish array of pointers with pointer to an array with conceptual points and syntaxes.	2	4	3	4

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

---

	b) Debug the following code which is depicting the concept of array of pointers and pointer to an array such that the output is as follows
	`Array of Pointers Output:`
	`Red`
	`Green`
	`Blue`
	
	`Pointer to Array Output:`
	`Cat`
	`Dog`
	`Cow`
	`#include <stdio.h>`
	`int main() {`
	`  char *names= {"Red", "Green", "Blue"};//Line 1`
	`  char arr[3][10] = {"Cat", "Dog", "Cow"};//Line 2`
	`  char *p[10] = arr;  //Line 3`
	`  printf("Array of Pointers Output:\n");`
	`  for(int i = 0; i < 3; i++) {`
	`  printf("%s\n", *names);  //Line 4`
	`  }`
	`  printf("\nPointer to Array Output:\n");`
	`  for(int i = 0; i < 3; i++) {`
	`  printf("%s\n", *p[i]);  ////Line 5`
	`  }`
	`  return 0;`
	`}`
		2	6		

5.	In a smart-home IoT energy monitoring system, a controller collects variable numbers of power-usage readings (in watts) from Wi-Fi enabled smart plugs connected to home appliances. The number of appliances can change at runtime, and additional devices may join the network later. To efficiently handle this dynamic data, write a C program that: a) Reads the initial number of appliances n. b) Allocates memory for storing n power readings using malloc(). c) Initializes another array of size n using calloc() to store corrected (noise-filtered) readings. d) Later, if k more devices are added, use realloc() to expand the readings array to size n+k and input the new readings. e) Display: Original readings, Corrected readings (copy original to corrected array), Final list of readings after using realloc(). f) Finally, free all dynamically allocated memory. Sample output:	2	10	3	5

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

```
Enter number of IoT appliances: 2
Enter power readings (in watts):
80
100

Enter number of new appliances: 1
Enter readings of new appliances:
75

Original Readings:
80 100

Corrected Readings (using calloc):
80 100

Final Readings after realloc():
80 100 75
```

****

Answer Key

SLOT: D2

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

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

Programme Name & Branch : B.Tech. & All
Course Code and Course Name : BACSE104 and Structured and Object-Oriented Programming
Date of Examination : 30th Jan, 2026
Exam Duration : 90 minutes **Maximum Marks: 50**

Q. No	Question	Marks

1.	#include <stdio.h>
	
	int main() {
	int n, calls;
	char name[30];
	float bill, extra;
	printf("Enter number of customers: ");
	scanf("%d", &n);
	printf("\nS.No\tName\t\tCalls\tBill\n");
	printf("---\n");
	for (int i = 1; i <= n; i++) {
	printf("\nEnter customer name: ");
	scanf("%s", name);
	printf("Enter number of calls: ");
	scanf("%d", &calls);
	
	// --------------------------
	// Bill calculation
	// --------------------------
	if (calls <= 200)
	bill = calls * 0.80;
	
	else if (calls <= 500)
	bill = (200 * 0.80) + (calls - 200) * 0.50;
	
	else
	bill = (200 * 0.80) + (300 * 0.50) + (calls - 500) * 0.30;
	
	// --------------------------
	// Minimum bill condition
	// --------------------------
	if (bill < 150)
	bill = 150;
	
	// --------------------------
	// Surcharge condition
	// --------------------------
	if (bill > 1000)
	bill = bill + (0.10 * bill);
		10

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

```
    // -------------------------
    // Print bill details
    // -------------------------
    printf("%d\t%s\t\t%d\t%.2f\n", i, name, calls, bill);
  }

  return 0;
}
```

2. Explain in detail the different storage classes available in C. Discuss how each storage class affects the scope, visibility, and lifetime of variables. — 10

Storage Class	Declaration	Storage	Default Initial Value	Scope	Lifetime
auto	Inside a function/block	Memory	Unpredictable	Within the function/block	Within the function/block
register	Inside a function/block	CPU Registers	Garbage	Within the function/block	Within the function/block
extern	Outside all functions	Memory	Zero	Entire the file and other files where the variable is declared as extern	program runtime
Static (local)	Inside a function/block	Memory	Zero	Within the function/block	program runtime
Static (global)	Outside all functions	Memory	Zero	Global	program runtime

3. — 10

```
#include <stdio.h>
#include <ctype.h>   // for tolower()
int main() {
  char str[200];
  int a = 0, e = 0, i_c = 0, o = 0, u = 0;
  int x;
  printf("Enter a string: ");
  fgets(str, sizeof(str), stdin);
  for (x = 0; str[x] != '\0'; x++) {
    char ch = tolower(str[x]);  // Make lowercase to ignore case sensitivity
    if (ch == 'a')     a++;
    else if (ch == 'e') e++;
    else if (ch == 'i') i_c++;
    else if (ch == 'o') o++;
    else if (ch == 'u') u++;
  }
  // Print only the vowels that appear at least once
  if (a > 0) printf("a: %d\n", a);
  if (e > 0) printf("e: %d\n", e);
  if (i_c > 0) printf("i: %d\n", i_c);
  if (o > 0) printf("o: %d\n", o);
  if (u > 0) printf("u: %d\n", u);
```

```
    return 0;
}
```

4. a) Distinguish array of pointers with pointer to an array with conceptual points and syntaxes. — 4

Feature	Array of Pointers	Pointer to an Array
Meaning	Array containing pointers	Pointer pointing to full array
Syntax	`int *p[5]`	`int (*p)[5]`
Number of pointers	Many	Only one
Memory	Non-contiguous	Contiguous block
Main use	Strings, jagged arrays	Array traversal, 2D arrays
Access	`p[i]` gives pointer	`(*p)[i]` gives element

b) Debug the following code which is depicting the concept of array of pointers and pointer to an array such that the output is as follows — 6

```
Array of Pointers Output:
Red
Green
Blue

Pointer to Array Output:
Cat
Dog
Cow
```

```
#include <stdio.h>
int main() {
   char *names[]= {"Red", "Green", "Blue"};//Line 1
   char arr[3][10] = {"Cat", "Dog", "Cow"};//Line 2
   char (*p)[10] = arr;    //Line 3
   printf("Array of Pointers Output:\n");
   for(int i = 0; i < 3; i++) {
      printf("%s\n", names[i]);   //Line 4
   }
   printf("\nPointer to Array Output:\n");
   for(int i = 0; i < 3; i++) {
      printf("%s\n", p[i]);     ////Line 5
   }
   return 0;
}
```

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

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

5.	In a smart-home IoT energy monitoring system, a controller collects variable numbers of power-usage readings (in watts) from Wi-Fi enabled smart plugs connected to home appliances.The number of appliances can change at runtime, and additional devices may join the network later. To efficiently handle this dynamic data, write a C program that:
	a) Reads the initial number of appliances n.
	b) Allocates memory for storing n power readings using malloc().
	c) Initializes another array of size n using calloc() to store corrected (noise-filtered) readings.
	d) Later, if k more devices are added, use realloc() to expand the readings array to size n+k and input the new readings.
	e) Display:
	Original readings,Corrected readings (copy original to corrected array), Final list of readings after using realloc()
	f) Finally, free all dynamically allocated memory.
	
	Ans:
	
	#include <stdio.h>
	
	#include <stdlib.h>
	
	int main() {
	
	int n, k, i;
	
	// ---------- Step 1: Read initial appliances ----------
	
	printf("Enter number of IoT appliances: ");
	
	scanf("%d", &n);
	
	// ---------- Step 2: Allocate using malloc() ----------
	
	int *readings = (int *)malloc(n * sizeof(int));
		10

```
if (readings == NULL) {
   printf("Memory allocation failed\n");
   return 0;
}
printf("Enter power readings (in watts):\n");
for (i = 0; i < n; i++)
  scanf("%d", &readings[i]);


// ---------------------------------
// Step 3: Allocate using calloc()
// ---------------------------------
int *corrected = (int *)calloc(n, sizeof(int));
if (corrected == NULL) {
   printf("Memory allocation failed\n");
   free(readings);
   return 0;
}


// Copy original to corrected array
for (i = 0; i < n; i++)
   corrected[i] = readings[i];
```

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

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

```
// ---------------------------------
// Step 4: Add more IoT devices (realloc)
// ---------------------------------
printf("\nEnter number of new appliances: ");
scanf("%d", &k);

readings = (int *)realloc(readings, (n + k) * sizeof(int));
if (readings == NULL) {
   printf("Reallocation failed\n");
   free(corrected);
   return 0;
}

printf("Enter readings of new appliances:\n");
for (i = n; i < n + k; i++)
   scanf("%d", &readings[i]);

// ---------------------------------
// Step 5: Display all results
// ---------------------------------
printf("\nOriginal Readings:\n");
for (i = 0; i < n; i++)
```

```
    printf("%d ", readings[i]);


  printf("\n\nCorrected Readings (using calloc):\n");
  for (i = 0; i < n; i++)
    printf("%d ", corrected[i]);


  printf("\n\nFinal Readings after realloc():\n");
  for (i = 0; i < n + k; i++)
    printf("%d ", readings[i]);


  // ---------------------------------
  // Step 6: Free memory
  // ---------------------------------
  free(readings);
  free(corrected);


  return 0;
}
```

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

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

```
Enter number of IoT appliances: 2
Enter power readings (in watts):
80
100

Enter number of new appliances: 1
Enter readings of new appliances:
75

Original Readings:
80 100

Corrected Readings (using calloc):
80 100

Final Readings after realloc():
80 100 75
```

****