

Date: 19/12/2025

Value Added Course Plan

Course Title: Elite C Programming and Smart Optimization

Duration: Jan 05 – Jan 10 (6 Days)

Mode: Theory + Lab (Hands-on Sessions)

Total Hours: 36 hours (6 hours/day)

Faculty: Jerobin Susanna J / Senior developer – Project Trainer - 6380388783

Software Requirement: Turbo C / Code::Blocks / GCC Compiler

Department: Information Technology

Venue: Kamaraj College of Engineering and Technology

Course Fee: ₹1,500/- per student

Number of Students: 37

Total Course Fee: ₹55,500/- (Rupees Fifty Five Thousand Five Hundred Only)

Course Objectives

1. To strengthen foundational and advanced concepts of C programming for problem-solving.
2. To develop proficiency in pointers, structures, and dynamic memory management.
3. To introduce efficient coding practices and smart optimization techniques in C.
4. To enhance logical thinking and debugging skills using structured programming approaches.
5. To prepare students for placement-oriented C programming and technical interviews.

Day-wise Course Plan

Day	Date	Topics (Theory)	Lab Session / Practical Exercises	Expected Outcome
Day 1	Jan 05	Pointers – Deep Dive: <ul style="list-style-type: none"> • Concept of memory and addresses • Pointer declaration & initialization • Pointer arithmetic 	<ol style="list-style-type: none"> 1. Print address and values using pointers. 2. Swapping two numbers using pointers. 3. Implement pointer to 	Students can define and use pointers effectively, manipulate data through memory references.

		<ul style="list-style-type: none"> • Pointer to pointer • Pointer and arrays • Function pointers 	function for addition and subtraction. 4. Array traversal using pointers.	
Day 2	Jan 06	Structures & Unions: <ul style="list-style-type: none"> • Defining structures • Nested structures • Structure with pointers • Array of structures • Union vs Structure comparison 	<ol style="list-style-type: none"> 1. Create structure for student record (name, roll, marks). 2. Access structure using pointer. 3. Demonstrate union memory sharing. 4. Store and display 10 employee records using array of structures. 	Students understand how to use structures for data organization and memory efficiency.
Day 3	Jan 07	File Handling in C: <ul style="list-style-type: none"> • File types (text/binary) • fopen(), fclose(), fgetc(), fputc(), fscanf(), fprintf() • File read/write modes • Error handling in files 	<ol style="list-style-type: none"> 1. Write data to a file and read back. 2. Copy one file to another using file I/O. 3. Append content to an existing file. 4. Count words and characters in a file. 	Students can handle external file input/output operations and manage file-based data processing.
Day 4	Jan 08	Modular Programming & Functions: <ul style="list-style-type: none"> • Function prototypes • Header files • Modular approach using multiple source files • Scope, static, extern variables 	<ol style="list-style-type: none"> 1. Develop a program using multiple .c and .h files. 2. Create custom header for math operations. 3. Use static and extern keywords to manage variable scope. 4. Reuse functions across modules. 	Students learn structured modular programming with reusable code and better code organization.

Day 5	Jan 09	<p>Linked List, Stack & Queue:</p> <ul style="list-style-type: none"> • Dynamic memory allocation (malloc, free) • Singly linked list operations (create, insert, delete, traverse) • Stack implementation using linked list • Queue implementation using arrays 	<ol style="list-style-type: none"> 1. Create linked list and display all elements. 2. Insert and delete nodes from linked list. 3. Implement stack using linked list (push, pop). 4. Implement circular queue using arrays. 	<p>Students can build and manipulate dynamic data structures and understand memory management.</p>
Day 6	Jan 10	<p>Code Optimization & Final Review:</p> <ul style="list-style-type: none"> • Common subexpression elimination • Loop optimization techniques • Constant folding • Inline functions and macro efficiency • Best coding practices • Performance comparison demo 	<ol style="list-style-type: none"> 1. Optimize existing programs for time and space. 2. Compare execution time before and after optimization. 3. Create mini project integrating multiple topics (Pointers + Structures + Files). 4. Evaluation & presentation of final optimized code. 	<p>Students understand performance optimization and can write efficient, modular C programs.</p>

Assessment Plan

Component	Weightage	Description
Daily Lab Exercises	40%	Hands-on lab tasks each day
Mini Project	40%	Final day integrated coding challenge
Attendance & Participation	10%	Interaction and consistency
Viva / Quiz	10%	Conceptual understanding

Expected Outcomes

By the end of this course, students will be able to:

- Understand advanced C concepts (pointers, structures, memory allocation)
- Work with files and modular programs efficiently
- Implement real-world data structures (linked list, stack, queue)
- Optimize code for better performance and readability
- Build small to medium C applications following modular design



info@lithiinfotech.in
www.lithiinfotech.in



No : 2/401, Subhiksha Complex,
1st Floor, Vasanth Nagar, Palanganatham,
Madurai 625 003.