SYLLABUS

PROGRAMMING LANGUAGES · COURSE HANDOUT 2026

C & C++

NTERS . MEMORY . STRUCTS
CLASSES . TEMPLATES . STL
. MOVE SEMANTICS . BUILD

DURATION	LEVEL	MODULES	MODE
70 Hours	**Beginner to Advanced**	**8**	**Classroom · Online · Hybrid**

01 COURSE OVERVIEW

A systems programming course covering C fundamentals and modern C++. C comes first for memory and pointer discipline, then C++ builds up through classes, RAII, templates, the STL and move semantics, with modern practice (smart pointers, algorithms, CMake) rather than legacy habits.

Learners implement data structures in C and a small application in modern C++.

WHO IT IS FOR

- Engineering students
- Embedded and systems developers
- Competitive programmers
- Developers needing performance control

PREREQUISITES

None, though prior programming exposure helps.

02 LEARNING OUTCOMES

01 Manage memory manually without leaks.

02 Work confidently with pointers and arrays.

03 Implement core data structures from scratch.

04 Design classes with correct construction and destruction.

05 Use templates and the STL effectively.

06 Apply RAII, references and move semantics.

07 Build multi-file projects with CMake and debug them.

03 MODULE STRUCTURE

8 MODULES

01 C FUNDAMENTALS

- Types and operators
- Functions
- Strings
- Preprocessor
- Control flow
- Arrays
- Input and output

LAB

C EXERCISES

02 POINTERS AND MEMORY

- Pointer arithmetic
- Stack vs heap
- Function pointers
- malloc and free
- Structs and unions
- Common bugs

LAB

DYNAMIC STRUCTURES

03 DATA STRUCTURES IN C

- Linked lists
- Trees
- Sorting and searching
- Stacks and queues
- Hash tables
- Complexity

LAB

STRUCTURE LIBRARY

04 C++ BASICS

- References
- Overloading
- Strings and streams
- Namespaces
- Default arguments
- auto and range-for

05 CLASSES AND RAII

- Constructors and destructors
- Operator overloading
- Smart pointers
- Copy semantics
- const correctness
- Rule of five

06 TEMPLATES AND STL

Function and class templates	Containers
Iterators	Algorithms
Lambdas	std::optional and variant

LAB

GENERIC UTILITIES

07 MODERN C++

Move semantics	Perfect forwarding
constexpr	Concurrency with std::thread
Chrono	Error handling strategies

08 TOOLING

Compilation and linking	CMake
Debugging with gdb	Sanitizers
Static analysis	Unit tests with Catch2

PROJECT

MODERN C++ APPLICATION

04 ASSESSMENT & CERTIFICATION

Module assignments and labs	**25%**
Mini projects	**20%**
Internal assessments	**15%**
Final project and review	**40%**

CERTIFICATION

Learners who complete all modules, submit the final project and clear the review receive a course completion certificate from Kaizen Infinities Private Limited. Project work is documented for the learner's portfolio, and interview preparation is included in the closing sessions.

05 CAREER OUTCOMES

ROLES THIS PROGRAMME PREPARES FOR

C/C++ DEVELOPER **EMBEDDED SOFTWARE ENGINEER** **SYSTEMS PROGRAMMER** **GAME PROGRAMMER** **FIRMWARE ENGINEER**

CALL US
+91 73051 33996

EMAIL US
apply@kaizeninfinities.com

VISIT OUR WEBSITE
www.kaizeninfinities.com