LAPT

LONDON ACADEMY OF PROFESSIONAL TRAINING

CERTIFICATE

ISO 9899 — C Programming Language Standard

ISO Certification Programme

LAPT — London Academy of Professional Training

Address 85 Great Portland Street, W1W 7LT, London, United Kingdom

Email info@lapt.org

Telephone +44 7513 283044

Web lapt.org

Reference IIT-COD-9899 • ISO IT & Related Technologies

UKPRN 10067351 • Registered in England and Wales, company 4984798

6	20	96	430	60%	Lifetime
SUBJECTS	CHAPTERS	CLASSES	TOTAL MARKS	PASS MARK	VALIDITY

AT A GLANCE

REFERENCE	**IIT-COD-9899**
AWARD	**Certificate**
ASSESSMENT	**430 marks — 258 to pass (60%)**
PREREQUISITES	**None**
VALIDITY	**Lifetime**
AWARDING BODY	**LAPT — London Academy of Professional Training**

CURRICULUM

1 Advanced C Programming Techniques

5 chapters · 30 classes **105 marks**

• 1 Memory Management and Dynamic Allocation in C — 6 classes

- › 1.1 Understand Memory Layout in C: Stack vs Heap
- › 1.2 Explore Dynamic Memory Allocation Functions: malloc, calloc, realloc
- › 1.3 Implement Memory Allocation Strategies: Best Practices
- › 1.4 Diagnose Memory Leaks: Tools and Techniques
- › 1.5 Master Pointer Arithmetic for Dynamic Data Structures
- › 1.6 Apply Dynamic Memory in Real-World Applications: Case Studies

• 2 Advanced Data Structures: Linked Lists, Trees, and Graphs — 6 classes

- › 2.1 Understand the Basics of Linked Lists
- › 2.2 Implement Single and Double Linked Lists
- › 2.3 Explore Operations on Linked Lists: Insertion and Deletion
- › 2.4 Introduction to Tree Data Structures
- › 2.5 Build and Traverse Binary Search Trees
- › 2.6 Analyze Graphs: Representations and Algorithms

• 3 Concurrency and Multithreading in C — 6 classes

- › 3.1 Understand Concepts of Concurrency and Multithreading
- › 3.2 Explore POSIX Threads for Multithreading in C
- › 3.3 Implement Thread Creation and Termination in C
- › 3.4 Manage Shared Resources with Mutexes in C
- › 3.5 Utilize Condition Variables for Thread Synchronization
- › 3.6 Design and Analyze a Multithreaded C Application

• 4 File I/O and Data Serialization Techniques — 6 classes

- › 4.1 Understand File I/O Concepts in C Programming
- › 4.2 Implement Basic File Operations with Standard Library Functions
- › 4.3 Explore File Modes and Their Impact on Data Access
- › 4.4 Introduce Data Serialization Techniques for C Structures
- › 4.5 Apply Advanced Serialization Methods with Binary Files
- › 4.6 Develop a Complete Project: File I/O and Data Serialization in Action

• 5 Interfacing with Hardware and System Programming — 6 classes

- › 5.1 Understanding Hardware Interfaces in C Programming
- › 5.2 Using Memory-Mapped I/O for Device Communication
- › 5.3 Implementing Interrupt Handling in C
- › 5.4 Writing Driver Code for Peripheral Devices
- › 5.5 Utilizing System Calls for Process Management
- › 5.6 Debugging and Testing Hardware Interface Code

2 C Programming Fundamentals

5 chapters · 30 classes **85 marks**

- **1 Getting Started with C: Syntax and Structure** — 6 classes
  - 1.1 Understand the Basic Syntax of C Programming
  - 1.2 Identify C Programming Structure: Functions and Main
  - 1.3 Explore Data Types and Variables in C
  - 1.4 Implement Operators and Expressions in C
  - 1.5 Utilize Control Structures: If Statements and Loops
  - 1.6 Create a Simple C Program: Putting It All Together
- **2 Data Types and Variables: The Building Blocks of C** — 6 classes
  - 2.1 Explore Basic Data Types in C
  - 2.2 Understand Variable Declaration and Initialization
  - 2.3 Analyze Scope and Lifetime of Variables
  - 2.4 Differentiate Between Primitive and Derived Data Types
  - 2.5 Practice Type Conversion and Type Safety
  - 2.6 Apply Variables in Simple C Programs
- **3 Control Flow: Making Decisions in C** — 6 classes
  - 3.1 Understand Conditional Statements: if Statements in C
  - 3.2 Implementing Multiple Conditions: Using if-else Statements
  - 3.3 Exploring Logical Operators: Using && and || in Conditions
  - 3.4 Simplifying Decisions: Utilizing the switch Statement
  - 3.5 Nesting Control Structures: Combining if-else and switch
  - 3.6 Practicing Control Flow: Creating a Decision-Making Program
- **4 Functions and Scope: Organizing Code in C** — 6 classes
  - 4.1 Define and Explain Functions in C
  - 4.2 Explore Function Declarations and Definitions
  - 4.3 Understand Function Parameters and Return Types
  - 4.4 Investigate Scope and Lifetime of Variables
  - 4.5 Apply Functions to Simplify Code Management
  - 4.6 Develop a Complete Program Using Functions
- **5 Arrays and Pointers: Advanced Data Management in C** — 6 classes
  - 5.1 Understand the Basics of Arrays in C
  - 5.2 Explore Array Initialization and Declaration Techniques
  - 5.3 Navigate Advanced Pointer Concepts and Their Usage
  - 5.4 Examine the Relationship Between Arrays and Pointers
  - 5.5 Implement Dynamic Memory Allocation with Pointers
  - 5.6 Apply Arrays and Pointers in Real-world C Programming Scenarios

3 ISO Standards Compliance

5 chapters · 30 classes **65 marks**

- **1 Understanding ISO 9899: An Overview of C Standards** — 6 classes
  - 1.1 Explore the Origins and Development of ISO 9899
  - 1.2 Identify Key Features and Principles of C Standards
  - 1.3 Compare ISO 9899 with Previous C Language Standards
  - 1.4 Analyze the Impact of Compliance on C Programming Practices
  - 1.5 Examine Common Compliance Challenges and Solutions
  - 1.6 Apply ISO 9899 Guidelines in a Practical C Coding Scenario
- **2 Key Principles of ISO Compliance in C Programming** — 6 classes
  - 2.1 Understand the Basics of ISO 9899 and Its Purpose
  - 2.2 Identify Key C Programming Standards in ISO Compliance
  - 2.3 Explore the Importance of Code Consistency in ISO Standards
  - 2.4 Apply Best Practices for Documentation in C Programming
  - 2.5 Assess Code Quality Through ISO Compliance Checklists
  - 2.6 Implement Continuous Improvement Strategies in C Development
- **3 Analyzing C Standard Library Functions in ISO 9899** — 6 classes
  - 3.1 Identify Key Functions in the C Standard Library
  - 3.2 Examine Function Parameters and Return Types
  - 3.3 Analyze Common C Standard Library Use Cases
  - 3.4 Explore Error Handling in C Standard Library Functions
  - 3.5 Compare Performance of Standard Functions
  - 3.6 Implement a C Program Utilizing Standard Library Functions
- **4 Compliance Testing and Verification Techniques** — 6 classes
  - 4.1 Define Compliance Testing and its Importance in ISO Standards
  - 4.2 Identify Key Verification Techniques for ISO 9899 Compliance
  - 4.3 Develop a Compliance Testing Checklist for C Programs
  - 4.4 Implement Automated Tools for Compliance Verification
  - 4.5 Evaluate Test Results and Identify Non-Compliance Issues
  - 4.6 Create a Compliance Improvement Action Plan Based on Findings
- **5 Best Practices for Maintaining ISO Compliance in Development** — 6 classes
  - 5.1 Understand ISO 9899 Standards and Their Importance
  - 5.2 Identify Key Compliance Requirements for Development Teams
  - 5.3 Implement Documentation Practices for ISO Compliance
  - 5.4 Adopt Version Control Systems to Support ISO Standards
  - 5.5 Establish a Continuous Improvement Process for Compliance
  - 5.6 Review and Audit Your Development Practices for Compliance

4 Project Management Strategies

5 chapters · 6 classes · 85 marks

- **1 Introduction to Project Management Principles in IT** — 6 classes
  - 1.1 Define the Key Principles of Project Management in IT
  - 1.2 Identify Roles and Responsibilities in IT Project Teams
  - 1.3 Explore Common Project Management Methodologies for IT
  - 1.4 Analyze Project Life Cycles in Information Technology
  - 1.5 Develop Effective Communication Strategies for IT Projects
  - 1.6 Apply Risk Management Techniques to IT Project Planning
- **2 Defining Project Scope and Objectives**
- **3 Risk Management Strategies in Software Development**
- **4 Agile Methodologies and Their Application**
- **5 Monitoring, Evaluation, and Project Closure Techniques**

5 Quality Assurance and Testing

0 chapters · 45 marks

6 Team Leadership in IT

0 chapters · 45 marks