

Introduction to C Programming | C Programming Fundamentals (TTMFC2100)

Explore ANSI C Language, Emphasizing Portability, Structured Design & Best Practices

www.triveratech.com

Course Snapshot

- **Course:** Introduction to C Programming | C Programming Fundamentals (TTMFC2100)
- **Duration:** 5 days
- **Skill Level:** Introductory
- **Audience:** This is a developer-focused course geared for new or newer programmers.
- **Hands-on Learning:** This course is approximately **50% hands-on**, combining expert lecture, real-world demonstrations and group discussions with machine-based practical programming labs and exercises. Student machines are required.
- **Language:** This course examples are in Java. This course is also available for Node.JS and .Net. Please inquire for details.
- **Delivery Options:** This course is available for **onsite private classroom presentation, live online virtual presentation**, or can be presented in a **flexible blended learning format** for combined onsite and remote attendees. Please also ask about our **Self-Paced / Video** or **QuickSkills / Short Course** options.
- **Public Schedule:** This course has active dates on our live-online open enrollment **Public Schedule**.
- **Customizable:** This course agenda, topics and labs can be further adjusted to target your specific training skills objectives, tools and learning goals. Please inquire for details.

Overview

Introduction to C Programming / C Programming Fundamentals is a skills-focused, hands-on training course that provides students with a comprehensive introduction to the ANSI C language, emphasizing portability and structured design. Students are introduced to all major language elements including fundamental data types, flow control, and standard function libraries.

Throughout the course, students will thoroughly explore string and character manipulation, dynamic memory allocation, standard I/O, macro definition, and the C runtime library. The course explains the use of aggregate structures, unions, and pointers early on so the students can practice extensively in the hands-on labs. Structured programming constructs and varargs functions are also covered. Emphasis is given to the processing of command line arguments and environment variables so students will be able to write flexible, user-friendly programs. The course also includes coverage of portability tips drawn from experienced programmers working in production environments. Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency.

Learning Objectives

This 'skills-centric' course is about **50% hands-on lab and 50% lecture** designed to train attendees in core C development skills, coupling the most current, effective techniques with the soundest coding practices. Students will explore a series of progressively advanced topics, where each topic consists of lecture, group discussion, practical hands-on lab exercises, and lab review.

Working within an engaging, hands-on learning environment, attendees will explore:

- Components of a C program
- Using the C preprocessor
- Using standard runtime libraries
- Using `make` to build programs
- Working with debugger utilities
- Using data types, storage classes and scope
- Using `typedef` to make code more readable and portable
- Using operators and expressions
- Working with conditional and looping constructs
- Initializing a pointer
- Accessing the value addressed by a pointer
- Returning the value of a function

- Declaring argument data types
- ANSI function prototype syntax
- Declaring and initializing arrays and multidimensional arrays
- Using Strings and character manipulation
- Declaring and instancing a structure
- Defining a union
- Accessing command line arguments and environment variables
- C runtime library standard I/O functions

Audience & Pre-Requisites

This is an **introductory-level** C programming course designed for attendees new to C, and new or newer to programming. Attendees should be comfortable working with the command line and should have a good foundational mathematics or logic skills

Enhanced Learning Services: Please also ask about our **Pre-Training Class OnRamp & Prep / Primer** offerings, **Skills Gap Assessment Services, Case Studies, Knowledge Check Quizzes, Skills Immersion Programs & Camps, Collaborative Mentoring Services and Extended Learning Support & Post Training** services.

Course Topics / Agenda

Please note that this list of topics is based on our standard course offering, evolved from typical industry uses and trends. We will work with you to tune this course and level of coverage to target the skills you need most. Course agenda, topics and labs are subject to adjust during live delivery in response to student skill levels and interests.

- 1. Overview of C**
 - Operating System Independence
 - Design Goals and Capabilities
 - Flavors of C
- 2. Compiler Directives and the C Preprocessor**
 - Compile-Time Directives
 - Use of typedef
 - C Preprocessor Syntax
- 3. Fundamental Data Types, Storage Classes, and Scope**
 - Fundamental Data Types and Qualifiers
 - Constants and Strings
 - Storage Classes
 - Scope and Block Structure
 - Scope and Data Hiding
 - Data Initialization
- 4. Pointers and Dynamic Allocation**
 - Advantages of Pointers
 - User of Pointers
 - Pointer and Address Arithmetic
 - Dynamic Storage Allocation
 - sizeof Operator
 - Double Indirection
- 5. Macros**
 - Functions vs. Inlining
- 6. Arrays**
 - Purpose of Arrays
 - Declaring an Array
 - Initializing an Array
 - Addressing Elements
 - Stepping Through an Array
 - Variable Size Arrays
 - Arrays of Pointers
 - Arrays of Strings
 - Passing an Array to a Function
 - Dynamic Memory Allocation
 - Multidimensional Arrays
- 7. Basic Formatted I/O**
 - Standard I/O Library
- 8. Program Debugging**
 - Problem Analysis
 - Instrumenting with printf
 - Instrumenting with ctrace
 - The Purpose of Debuggers
 - How Not to Use Debuggers
 - Symbolic Debuggers
- 9. Operators and Expressions**
 - Arithmetic, Logical, and Bit Operators
 - Precedence and Associativity
 - Assignment and Casting
 - The Conditional Operator
- 10. Flow Control Constructs**
 - Conditional Constructs: if, switch
 - Looping Constructs: while, do, for
 - Programming Style
- 11. Functions (Subroutines)**
 - Purpose of Functions
 - Functions vs. Inlining
 - Automatic Variables
 - The Argument Stack

- Passing By Value
 - Passing By Reference
 - Declaring External Functions
 - Function Prototyping
 - ANSI Prototyping
 - The `_NO_PROTO` Compiler Symbol
 - Varargs Functions
 - Passing a Function as an Argument
 - Designing Functions for Reusability
 - Calling a Function from Another Language
 - Returning a Dynamically Allocated Value Using Double Indirection
 - Casting the Return Value of a Function
 - Recursion and Reentrancy
- 12. Structures**
- Purpose of Structures
 - Defining and Declaring Structures
 - Accessing Members
- Pointers to Structures
 - Dynamic Memory Allocation
 - Passing a Structure to a Function
 - As a Pointer
 - Passing the Actual Structure
- 13. Advanced Structures and Unions**
- Nested Structures
 - Arrays of Structures
 - Bit Fields
 - Unions
 - Linked Lists
- 14. C Runtime Library Standard Functions**
- Character I/O
 - Unformatted File I/O
 - Formatted File I/O
 - Math Functions
 - Miscellaneous Functions
- 15. Strings and Character Manipulation**
- Strings as Character Arrays
 - String Library Functions
- Reading and Writing Strings
- 16. Accessing Command Line Arguments and Environment Symbols**
- `argc` and `argv`
 - Parsing Command Line Options
 - Accessing the Environment Array
- 17. Structured Programming**
- Structuring Code for Quality, Reliability, Maintainability
 - Designing for Modularity and Reusability
- 18. Advanced Programming Consideration**
- Writing Portable Code
 - Use of Macros
 - ANSI C Limits
 - Feature Test Macros
 - Client/Server Design
 - [Performance Considerations](#)

Need different skills or topics? If your team requires different topics or tools, additional skills or custom approach, this course may be further adjusted to accommodate. We offer additional C, COBOL, structured programming, mainframe and other related courses which may be blended with this course for a track that best suits your development objectives. Our team will collaborate with you to understand your needs and will target the course to focus on your specific learning objectives and goals.

Student Materials: Each student will receive a **Student Guide** with course notes, code samples, software tutorials, diagrams and related reference materials and links (as applicable). Our courses also include step by step hands-on lab instructions and solutions, clearly illustrated for users to complete hands-on work in class, and to revisit to review or refresh skills at any time. Students will also receive the project files (or code, if applicable) and solutions required for the hands-on work.

Classroom Setup Made Simple: Our dedicated tech team will work with you to **ensure your classroom and lab environment is setup, tested and ready to go** well in advance of the course delivery date, ensuring a smooth start to class and seamless hands-on experience for your students. We offer several flexible student machine setup options including **guided manual set up** for simple installation directly on student machines, or **cloud based / remote hosted lab solutions** where students can log in to a complete separate lab environment minus any installations, or we can supply **complete turn-key, pre-loaded equipment** to bring ready-to-go student machines to your facility. Please inquire for details, options and pricing.

For More Information

Need dedicated training? All courses can be presented **onsite** or **online**, or in a **combined / flex / blended learning format**, tailored to target your specific audience, needs and learning goals. We also offer focused, flexible **short courses, self-paced learning options, recorded sessions** and more. We train beginner to advanced skills in all areas we cover, and offer **New Hire / Cohort Training, Boot Camps, Skills Immersion Programs, Reskilling Programs, Skills Migration & Transition Programs**, and more. We collaborate with you to ensure all courses are truly targeted to meet your specific needs and learning skills, maximizing your valuable training time, as well as your important budget.

