Specialized - Mastering C Programming

Code:	C-PROG
Length:	5 days
URL:	View Online

Mastering C Programming is an intermediate to advanced level C programming course provides experienced C Language Programmers with additional insights into the language. The student is given an in-depth understanding of the use of pointers, arrays, and structures in C Language, and is prepared to use these language facilities to write readable, portable, and efficient C Language Programs.

Skills Gained

- Evaluate simple and complex expressions.
- Effectively use expressions involving side effects.
- Describe and use all C Language basic and derived data types.
- Describe the attributes and results of all C Language operators.
- Explain the scope and storage class of C Language objects.
- Declare and use pointer data types and multi-levels of indirection.
- Describe pointer arithmetic and use it to access array elements.
- Declare and use single and multi-dimensional arrays.
- Describe the difference between an array of pointers and a multi-dimensional array.
- Reference single and multi-dimensional array elements using both pointers and array subscripting.

Who Can Benefit

Experienced C Language Applications Programmers, and Systems Programmers.

Prerequisites

At least six (6) months of C Language Programming experience.

Course Details

I. Introduction

- Evaluating Expressions
- Expression Side Effects
- Objects and Data Types
- Lvalues and Constants

II. Data Types, Constants, and Variables

- Basic Data Types
- Constant Data Types

- Variable Data Types
- Derived Data Types
- Data Type Renaming - typedef

III. Scope and Storage Class

- Internal Scope
- External Scope
- Extending Scope
- Storage Class
- Automatic
- External
- Static
- Register

IV. Operators And Operands

- Operator Attributes
- Operator Action
- Number of Operands
- Precedence Level
- Associativity Direction
- Operand Attributes
- Data Type Restrictions
- Lvalue Requirements
- Expression Evaluation Result
- Data Type
- Value
- Lvalue
- Side Effects

V. Arrays and Pointers

- Multi Dimensional Arrays
- Pointer - Array Relationship
- Pointer Arithmetic
- Valid Pointer Operations
- Multiple Pointers to an Array
- Array Subscripting vs. Pointers
- Arrays of Pointers
- Multiple Levels of Indirection

VI. Structures and Pointers

- Structure Review

- Structure Definition
- Structure Declaration
- Structure Holes
- Nested Structures
- Arrays of Structures
- Pointers to Structures
- Structures Containing Pointers
- Structure Variations
- Unions
- Bit Fields
- Enumerations

VII. Functions and Pointers

- Function Declaration
- Old Style
- Function Prototypes (ANSI)
- Function Arguments and Parameters
- Basic Data Types
- Pointers to Basic Data Types
- Single and Multi-Dimensional Arrays
- Structures and Pointers to Structures
- Function Return Types
- void Data Type Functions
- Basic Data Type Functions
- Pointer to Data Type Functions
- Structure Data Type Functions
- Pointers to Functions
- Declaring Pointers to Functions
- Pointers to Functions as Formal Parameter
- Arrays of Pointers to Functions
- Functions Returning Pointers to Functions

VIII. Dynamic Memory Allocation & Linked Lists

- Dynamic Memory Allocation
- Dynamic Memory Allocation Concepts
- Library Function: malloc
- Library Function: free
- Linked Lists
- Linked List Concepts
- Linear Linked List
- Bi-Directional Linked List

- Linked List Conclusions