

C Programming Language

Course No.

2300

Description

This course is intended to provide experienced programmers with an understanding of the basic elements of the C programming language. The course uses extensive hands-on workshops to develop a complete application in C, to reinforce the topics covered in each chapter. By the end of the course, students will have developed and debugged C programs that use branching, loops, functions, arrays, pointers and structures, and will have used file I/O functions to read and write data files to disk.

Audience

Programmers, systems analysts and managers who will be using the C programming language for development.

Prerequisites

Programming experience with at least one other programming language such as Basic, COBOL, or Pascal is helpful but not required

Objectives

- Develop applications that use simple variables for screen output and data capture from the keyboard.
- Use branching to control the flow of program statements.
- Use loops to perform one or more statements a number of times.
- Write applications that use functions to perform structured programming techniques.
- Understand and use arrays and strings in C programs.
- Write programs that use pointers for fundamental operations. Use structures to create and manage records of data.
- Perform file I/O operations such as reading and writing disk files

Major Topics

- Introduction to C
- Building Blocks
- Operators, Expressions, and Statements
- Branching
- Loops
- Functions
- Arrays and Strings
- Pointer Concepts
- Structures
- File I/O

Format

Lecture and Hand-On Format

Duration

1 day



Course Contents

1. Introduction to C

- What is C?
- Features of C
- History of C
- A Simple C Program
- Style in C
- Using an Editor
- Compiling the Program
- The Program Development Cycle

2. Building Blocks

- Overview
- Basis Data Types
- Variables
- Strings
- Formatted Output With printf()
- The printf() Format Specifics
- Conversion Characters
- Field Width
- Precision
- Escape Characters
- Receiving Input with scanf()
- Using scanf()
- Problems with scanf ()
- Character Functions
- String Input

3. Operators, Expressions and Statements

- Overview
- Variables
- Identifiers
- Reserved Words
- Declaring Variables
- Common Errors Declaring Variables
- Constants
- Operators
- Integer Arithmetic
- Increment/Decrement Operators
- Mixing Data Types
- Casting
- sizeof()
- Assignment
- Expressions
- Precedence

- Statements
- Putting It All Together

4. Branching

- Overview
- Relational Operators
- Equality vs. Assignment
- True and false
- Logical Operators
- Logical AND
- Logical OR
- Conditional Operator
- The if Statement
- Common Errors with if Statements
- The if-else if Statement
- Common Errors
- Nested if Statements
- The switch Statement
- Using the switch statement

5. Loops

- Overview
- Looping and Iteration
- The for Loop
- Operation of a for Loop
- Elements of a for Loop
- Multiple Statements in for Loops
- Common Errors in for Loops
- while Loops
- Combining Increment and Text Statements
- Common Errors in While Loops
- The do-while Loop
- Infinite Loops
- break
- continue
- Using continue
- Nested Loops
- Summary
- Debugging

6. Functions

- Overview
- Using Functions
- Simple Functions



- Declarations vs. Definitions
- Passing Values to Functions
- Passing Arguments by Value
- Return Type
- Returning a Value
- Macros
- Problems with Macros
- Visibility and Lifetime
- Automatic Variables
- Static Variables
- Using Static Variables
- External Variables
- Declarations vs. Definition
- Storage Class
- Setting a Program List
- Writing Large Programs

7. Arrays and Strings

- Overview
- What is an Array?
- Declaring an Array
- Initializing an Array
- Accessing Array Elements
- Exceeding Array Bounds
- Two-Dimensional Arrays
- Arrays and Memory
- Array Restrictions
- Passing Arrays to Functions
- Value vs. Reference
- Strings
- Array of Strings
- String Functions
- Using String Functions

8. Pointer Concepts

- Overview
- Using Pointers
- What is a Pointer?
- Three Steps to Using Pointers
- The Indirection Operator
- Pointers and Data Types
- A simple Pointer Example

- Common Pointer Problems
- Pointer Operations
- Pointer Arithmetic
- Using Pointer Arithmetic
- Call by Value
- Call by Reference
- Returning Multiple Values
- Allocating Memory
- Returning a Pointer from a Function
- argv and argc

9. Structures

- Overview
- What is A Structure?
- Structure Members
- Defining a Structure Template
- Using the Structure Definition
- Declaring a Structure Variable
- Accessing Structure Members
- Arrays of Structures
- Assigning Structures
- Pointers to Structures
- Passing Structures to Functions
- Passing Structures by Reference
- Returning Structures from Functions
- Application: Linked List
- Using malloc() with Structures

10. File I/O

- Overview
- Files
- Opening a File
- File Access
- Text and Binary Modes
- Characters and Strings I/O
- String Output
- String Input
- Block or Record I/O
- fwrite()
- fread()
- Sequential and Random Access Custom Tags and JSTL

