

C PROGRAMMING

Current Version: c-3.1.2

Course Length: 5 days

Course Description: Students will develop the ability to design and write programs in the C programming language.

Audience: Programmers new to the ANSI C language.

Prerequisites: Programming skill in a language such as Pascal, COBOL, BASIC, or assembler.

Course Contents

Introduction to C

- What is C ?
- Features of C
- Why Program in C ?
- History of C
- Current Status and Future

An Overview of C

- The First Program (hello.c)
- How to Compile and Run a C Program
- An Arithmetic Program (roof.c)
- Execution Flow Control (mph.c)
- The for Loop
- The for Loop - Diagram
- Character I/O
- A File Copier Program (cp2.c)
- A Character Counter (wc2.c)
- A Look at Arrays
- Stock Values (stock1.c)
- The char Data Type
- Strings (Character Arrays)

- A String Copy Program (stringcp.c)
- A Look at Functions
- A Functional Program (func1.c)
- A Review of printf()

Data Types and Variables

- Fundamental Data Types
- Data Type Values and Sizes
- Variable Declarations
- Variable Names
- Constants
- Character Constants
- String Constants

Operators and Expressions

- What are Expressions?
- Arithmetic Operators
- Relational Operators
- Assignment Operator
- Expressions Have Resulting Values
- True and False
- Logical Operators
- Increment and Decrement Operators

The C Preprocessor

- Symbolic Constants
- Macro Substitution
- File Inclusion

Pointers and Arrays

- What is a Pointer?
- Pointer Operators
- Example: Pointers
- Why Use Pointers?
- Arrays
- Arrays (a Picture)

- The & Operator
- Pointers and Arrays
- Pointer Arithmetic
- Pointer Arithmetic (a Picture)
- Arrays and Pointers
- Array Names are Constant Pointers
- Passing Arrays to Functions
- Initializing Arrays

Advanced Pointers

- Pointer Initialization
- Command-Line Arguments
- Strings and Character Pointers
- Arrays of Pointers
- Command-Line Arguments
- Access Through Pointers
- Functions and Pointers
- Example: Functions and Pointers

Structures

- Structures
- Comparison of Structures and Arrays
- Structure Definitions
- Structure Declarations
- Structure Parameter Passing by Reference
- Pointers to Structures
- Structure Parameter Passing Again
- Arrays of Structures
- The malloc Routine (++ and --)
- Increment and Decrement Operators: Examples
- 'Operate-Assign' Operators (+=, *=, ...)
- Conditional Expression
- Operator Precedence
- Precedence and Order of Evaluation
- Evaluation of Logical Operators

- Type Conversions
- The Cast Operator
- Bitwise Logical Operators

Control Flow

- Statements
- if - else
- if() - else if()
- switch()
- while()
- do - while()
- for()
- The for Loop - Diagram
- Example: for() Loop
- Another Example: for() Loop
- The break Statement
- The continue Statement

Functions

- What is a Function?
- Example: findbig3()
- Why Use Functions?
- Anatomy of a Function
- Example: find_big_int()
- Arguments Passed by Value
- Addresses of Arguments Can Be Passed
- A Picture of Addresses and Values
- When to Use the Return Statement
- Returning Non-Integer Values
- Functions in Multiple Source Files
- A Simple make File
- The Concept of Variable Scope
- Automatic Variables
- Global (External) Variables
- Static Variables

- External Static Variables

Appendix - File I/O in C

- File Streams
- Predefined Streams
- The fprintf Function
- The fscanf Function
- fscanf() Examples
- The fputs and fgets Functions
- The fwrite and fread Functions
- System I/O