

C-LANGUAGE CURRICULAM

Duration: 2 Months.

1. Introducing C

- 1.1 History of C
 - Origin
 - Standardization
 - C-Based Languages
- 1.2 Strengths and Weaknesses Of C
 - Strengths
 - Weaknesses
 - Effective Use of C

2. C Fundamentals

- 2.1 Writing Simple Program
 - Program: Printing a PUN
 - Compiling and Linking
 - Integrated Development Environment
- 2.2 The General Form Of Simple Program
 - Directives
 - Functions
 - Statements
 - Printing Strings
- 2.3 Comments
- 2.4 Variables and Assignment
 - Types
 - Declarations
 - Assignment
 - Printing the value of a Variable
 - PROGRAM: Computing Dimensional Weight of a Box
 - Initialization
 - Printing Expressions
- 2.5 Reading Input
- 2.6 Defining Names for Constants
- 2.7 Identifiers Keywords
- 2.8 Layout of a C Program

3. Formatted INPUT/OUTPUT

- 3.1 The printf Function
 - Conversion Specifications
 - Escape Sequence
- 3.2 The scanf Function
 - How scanf works
 - Ordinary characters in format strings
 - Confusing printf with scanf

4. EXPRESSIONS

- 4.1 Arithmetic Operators
 - Operator Precedence and Associativity
- 4.2 Assignment Operators

- Simple Assignments
- Lvalues
- Compound Assignment
- 4.3 Increment and Decrement Operators
- 4.4 Expression Evaluation
 - Order of Sub Expression Evaluation
- 4.5 Expression Statements

5. Selection Statements

- 5.1 Logical Expressions
 - Relational Operators
 - Equality Operators
 - Logical Operators
- 5.2 if Statement
 - Compound Statement
 - else clause
 - Cascaded if Statement
 - The Dangling else Problem
 - Conditional Expressions
 - Boolean values in C89
 - Boolean values in C99
- 5.3 The switch Statement
 - The Role of break statement

6. LOOPS

- 6.1 The while Statement
 - Infinite Loops
- 6.2 The do Statement
- 6.3 The for Statement
 - for Statement Idioms
 - Omitting Expressions in a for Statement
 - for statements in C99
 - The comma Operator
- 6.4 Exiting from Loop
 - The break Statement
 - The continue Statement
 - The goto Statement
- 6.5 The Null Statement

7. Basic Data Types

- 7.1 Integer Type
 - Integer Types in C99
 - Integer Constants
 - Integer Constants in C99
 - Integer Overflow
 - Reading and Writing Integers
- 7.2 Floating Types
 - Floating Constants
 - Reading and Writing Floating-point Numbers
- 7.3 Character Types
 - Operations on Characters

- Signed and Un Signed Characters
- Arithmetic Types
- Escape Sequence
- Character-Handling Functions
- Reading and Writing Characters using scanf and printf
- Reading and Writing Characters using getchar and putchar

7.4 Type Conversion

- The Usual Arithmetic Conversions
- Conversion during Assignment
- Implicit Conversions in C99
- Casting

7.5 Type Definitions

- Advantages of Type Definition
- Type Definitions and Portability

7.6 The sizeof Operator

8. ARRAYS

8.1 One-Dimensional Arrays

- Arrays Subscripting
- Array Initialization
- Designated Initializers
- Using sizeof operators with Arrays
- PROGRAM: Computing Interest

8.2 Multi Dimensional Arrays

- Initializing a Multi Dimensional Array
- Constant Arrays

8.3 Variable Length Arrays

9. FUNCTIONS

9.1 Defining and Calling Functions

- Function Definitions
- Function Calls

9.2 Function Declarations

9.3 Arguments

- Argument Conversions
- Array Arguments
- Variable Length Array Parameters
- Using static in Array Parameter Declarations
- Compound Literals

9.4 The return Statements

9.5 Program Termination

- The exit function

9.6 Recursion

- The Quick Sort Algorithm

10 Program Organization

10.1 Local Variables

- Static Local Variables
- Parameters

10.2 External Variables

10.3 Blocks

- 10.4 Scope
- 10.5 Organizing a C Program

11 POINTERS

- 11.1 Pointer Variables
 - Declaring Pointer Variables
- 11.2 The Address and Indirect Operators
 - The Address Operator
 - The Indirection Operator
- 11.3 Pointer Assignment
- 11.4 Pointers as Arguments
- 11.5 pointers as Return Values

12 POINTERS AND ARRAYS

- 12.1 Pointer Arithmetic
 - Adding an Integer to Pointer
 - Subtracting an Integer from a Pointer
 - Subtracting one pointer from another
 - Comparing Pointers
 - Pointers to Compound Literals
- 12.2 Using Pointers for ARRAY processing
 - Combining * and ++ Operators
- 12.3 Using Array Name as Pointers
 - Array Arguments
 - Using a Pointer as ARRAY name

13 STRINGS

- 13.1 String Literals
 - Escape Sequence in String Literals
 - Continuing a String Literal
 - How String Literals are Stored
 - Operations on String Literals
 - String Literals and Character Constants
- 13.2 String Variables
 - Initializing a string Variables
 - Character Arrays Versus Character Pointers
- 13.3 Reading and Writing Strings
 - Writing Strings Using printf and puts
 - Reading Strings Using scanf and gets
 - Reading Strings Character by Character
- 13.4 Accessing Characters in a String
- 13.5 Using C String Library
 - The strcpy (string copy) Function
 - The strlen (string Length) Function
 - The strcat (string concatenation) Function
 - The strcmp (String Comparison) Function
- 13.6 String Idioms
 - Searching for the End of a String
 - Copying a String
- 13.7 Arrays of Strings
 - Command-Line Arguments

14 STRUCTURES, UNIONS, AND ENUMERATIONS

14.1 Structure Variables

- Declaring Structure Variables
- Initializing Structure Variables
- Designated Initializers
- Operations on structures

14.2 Structure Types

- Declaring a Structure Tag
- Defining a Structure Type
- Structures as Arguments and Return Values
- Compound Literals

14.3 Nested Arrays and Structures

- Nested Structures
- Arrays of Structures
- Initializing an Array of Structures

14.4 Unions

- Using Unions
- Using Unions to Build Mixed Data Structures
- Adding a Tag field to a Union

14.5 Enumerations

- Enumeration Tags and Type Names
- Enumerations as Integers
- Using Enumerations to Declare “Tag Fields”.