C Programming

Duration: 35hrs

Topics:

1. Introduction to C
   1.1. Hello World Program - Compilation and Execution on Command Line
   1.2. Object File
   1.3. Compiler – Language Translator
   1.4. Linker
   1.5. High-Level Language
   1.6. Low-Level Language
   1.7. Machine Language

2. Values, Types, Variables, Expression, Operators
   2.1. Entry Point Function : main
   2.2. Value
   2.3. Variables
   2.4. Data Types
   2.5. Constants
   2.6. Escape Character
   2.7. Basic Input / Output
      2.7.1. printf
      2.7.2. scanf
   2.8. Expression
      2.8.1. Operand
      2.8.2. Operator
      2.8.3. Sub-expression
   2.9. Operators
      2.9.1. Arithmetic operators
      2.9.2. Relational operators
      2.9.3. Logical operators
      2.9.4. Unary Increment and Decrement operators
      2.9.5. Bitwise operators
      2.9.6. Ternary/Conditional Operator
      2.9.7. Compound Assignment operators
      2.9.8. The sizeof operator
      2.9.9. Precedence and order of evaluation
   2.10. Variable Initialization
   2.11. Variable Assignment

2.12. Difference between Initialization and Assignment
2.13. Mention that references are not possible in C
2.14. Variable Declaration : Go Right Go Left Rule

3. Control Flow
   3.1. Statement
   3.2. Block
   3.3. Scope
   3.4. Selection or Decision Control Statements
      3.4.1. if statement
      3.4.2. Nested conditional constructs
      3.4.3. switch statement

   3.5. Loops
      3.5.1. while
      3.5.2. do...while
      3.5.3. for
      3.5.4. Loop Interruption
         3.5.4.1. break statement
         3.5.4.2. continue statement
      3.5.5. exit function

4. typedef and Enumerations
   4.1. typedef
   4.2. Enumerations

5. Type Casting and Conversion
   5.1. What is Type Casting?

6. Functions
   6.1. Fundamentals of functions
   6.2. Function declaration and prototype
   6.3. Function definition
   6.4. Function call
   6.5. Caller and Callee
   6.6. Call by value
   6.7. The return statement
   6.8. void Data Type
   6.9. Function arguments
      6.9.1. Passing Arguments to a Function
   6.10. Scope Of Variables

6.11. Storage Classes
6.12. Automatic Variables
6.13. Local Variables
6.14. Static Variables
6.15. External Variables
6.16. Recursion
6.16.1. Recursive Function
6.17. Function Call Stack
6.18. Function Overloading not possible in C.
6.19. Variable Argument List
6.20. How arguments are passed from caller to callee?

7. Arrays
7.1. Definition
7.2. Declaration of Single Dimensional Array
7.3. Initialization of Single Dimensional Array
7.4. Array elements in memory
7.5. Multidimensional Arrays
7.5.1. Declaration of multi-dimensional arrays
7.5.2. Initialization of two-dimensional arrays
7.5.3. Memory Representation of Two-dimensional Arrays

8. Pointers
8.1. What is a pointer variable?
8.2. Address and Dereferencing (& and *) Operators
8.3. Pointer type Declaration
8.4. Pointer Assignment
8.5. Pointer Initialization
8.6. Pointer Arithmetic
8.7. Pointer Comparison
8.8. Pointers and Functions
8.8.1. Pointers to Functions
8.9. Functions returning Pointers
8.10. Pointers and Arrays
8.10.1. Pointer to Array
8.10.2. Arrays of Pointers
8.10.3. Pointers to Pointers

9. Function Revisited
9.1. Call by Value

9.2. Call by Pointer

10. Character and String
    - 10.1. Character Data Type
    - 10.2. Null Character
    - 10.3. ASCII Character Set
    - 10.4. Understanding strings
    - 10.5. Initializing Character Arrays
    - 10.6. Difference between an array and string.
    - 10.7. Writing string length, string copy function.
    - 10.8. Standard Library String Functions
        - 10.8.1. strcat ()
        - 10.8.2. strcmp ()
        - 10.8.3. strcpy()
        - 10.8.4. strlen()

11. Dynamic Memory Allocation
    - 11.1. Need for Dynamic Memory Allocation
    - 11.2. Stack Memory and Heap Memory
    - 11.3. sizeof operator revisited
    - 11.4. Functions
        - 11.4.1. malloc
        - 11.4.2. free
        - 11.4.3. calloc
        - 11.4.4. realloc

12. C Runtime Library and Operating System Call
    - 12.1. What is C Runtime Library?
    - 12.2. What is System Call?
    - 12.3. Difference between a Standard library and System Call.
    - 12.4. What is an API?

13. Static Variables and Static Functions Revisited
    - 13.1. Static Variable
    - 13.2. Static Function

14. The Preprocessor
    - 14.1. Macro substitution
    - 14.2. Macros with arguments
    - 14.3. Nesting of Macros
    - 14.4. Undefining a Macro

14.5. File Inclusion
14.5.1. Difference between including the files using < > and " "
14.5.2. Usage of INCLUDE Environment Variable.
14.6. Conditional Compilation
14.7. Defining Macros on Command Line using /D option
14.8. __LINE__ and __DATE

15. File Handling
15.1. Unformatted high-level disk I/O functions
15.2. File handling functions
15.2.1. fopen
15.2.2. fclose
15.2.3. getc
15.2.4. putc
15.2.5. fgets
15.2.6. fputs
15.2.7. fprintf
15.2.8. fscanf
15.2.9. fread
15.2.10. fwrite
15.2.11. feof
15.2.12. ferror
15.2.13. fseek
15.2.14. ftell
15.2.15. rewind

16. Structures and Union
16.1. Basics of Structures
16.1.1. Declaration of Individual Members of a Structure
16.2. Structure Variables
16.3. Structure Initialization
16.4. Accessing Structure Members
16.5. Nested Structures
16.6. Structures and Arrays
16.6.1. Arrays of Structures
16.6.2. Arrays within Structures
16.7. Structures and Pointers
16.7.1. Pointers to Structures
16.7.2. Structures Containing Pointers
16.8. Structures and Functions
16.8.1. Structures as Function Arguments

16.8.2. Structures as Function Values
16.9. Unions
16.9.1. Operations on a Union
16.9.2. Differences between Structures and Unions