

Introduction to Programming Using C

Course Outline:

Part I: Jumping Right In

1 What Is C Programming, and Why Should I Care?

- What Is a Program?
- What You Need to Write C Programs
- The Programming Process
- Using C

2 Writing Your First C Program

- A Down-and-Dirty Chunk of Code
- The main() Function
- Kinds of Data
- Characters and C
- Numbers in C
- Wrapping Things Up with Another Example Program

3 What Does This Do? Clarifying Your Code with Comments

- Commenting on Your Code
- Specifying Comments
- Whitespace
- A Second Style for Your Comments

4 Your World Premiere—Putting Your Program's Results

- Up on the Screen
- How to Use printf()
- The Format of printf()
- Printing Strings
- Escape Sequences
- Conversion Characters
- Putting It All Together with a Code Example

5 Adding Variables to Your Programs

- Kinds of Variables

- Naming Variables
- Defining Variables
- Storing Data in Variables

6 Adding Words to Your Programs

- Understanding the String Terminator
- The Length of Strings
- Character Arrays: Lists of Characters
- Initializing Strings

7 Making Your Programs More Powerful with #include and #define

- Including Files
- Placing #include Directives
- Defining Constants
- Building a Header File and Program

8 Interacting with Users

- Looking at scanf()
- Prompting for scanf
- Problems with scanf()

Part II: Putting C to Work for You with Operators and Expressions

9 Crunching the Numbers—Letting C Handle Math for You

- Basic Arithmetic
- Order of Operators
- Break the Rules with Parentheses
- Assignments Everywhere

10 Powering Up Your Variables with Assignments and Expressions

- Compound Assignment
- Watch That Order!
- Typecasting: Hollywood Could Take Lessons from C

11 The Fork in the Road—Testing Data to Pick a Path

- Testing Data
- Using if
- Otherwise...: Using else

12 Juggling Several Choices with Logical Operators

- Getting Logical
- Avoiding the Negative

- The Order of Logical Operators

13 A Bigger Bag of Tricks—Some More Operators for Your Programs

- Goodbye if...else; Hello, Conditional
- The Small-Change Operators: ++ and --
- Sizing Up the Situation

Part III: Fleshing Out Your Programs

14 Code Repeat—Using Loops to Save Time and Effort

- while We Repeat
- Using while
- Using do...while

15 Looking for Another Way to Create Loops

- for Repeat's Sake!
- Working with for

16 Breaking in and out of Looped Code

- Take a break
- Let's continue Working

17 Making the case for the switch Statement

- Making the switch
- break and switch
- Efficiency Considerations

18 Increasing Your Program's Output (and Input)

- putchar() and getchar()
- The Newline Consideration
- A Little Faster: getch()

19 Getting More from Your Strings

- Character-Testing Functions
- Is the Case Correct?
- Case-Changing Functions
- String Functions

20 Advanced Math (for the Computer, Not You!)

- Practicing Your Math
- Doing More Conversions

- Getting into Trig and Other Really Hard Stuff
- Getting Random

Part IV: Managing Data with Your C Programs

21 Dealing with Arrays

- Reviewing Arrays
- Putting Values in Arrays

22 Searching Arrays

- Filling Arrays
- Finders, Keepers

23 Alphabetizing and Arranging Your Data

- Putting Your House in Order: Sorting
- Faster Searches

24 Solving the Mystery of Pointers

- Memory Addresses
- Defining Pointer Variables
- Using the Dereferencing *

25 Arrays and Pointers

- Array Names Are Pointers
- Getting Down in the List
- Characters and Pointers
- Be Careful with Lengths
- Arrays of Pointers

26 Maximizing Your Computer's Memory

- Thinking of the Heap
- But Why Do I Need the Heap?
- How Do I Allocate the Heap?
- If There's Not Enough Heap Memory
- Freeing Heap Memory
- Multiple Allocations

27 Setting Up Your Data with Structures

- Defining a Structure
- Putting Data in Structure Variables

Part V: Files and Functions

28 Saving Sequential Files to Your Computer

- Disk Files
- Opening a File
- Using Sequential Files

29 Saving Random Files to Your Computer

- Opening Random Files
- Moving Around in a File

30 Organizing Your Programs with Functions

- Form Follows C Functions
- Local or Global

31 Passing Variables to Your Functions

- Passing Arguments
- Methods of Passing Arguments
- Passing by Value
- Passing by Address

32 Returning Data from Your Functions

- Returning Values
- The return Data Type
- One Last Step: Prototype