

# **Programming in C - Fundamentals**

## **1. Getting Started**

Whence C?

Why C?

Whither C?

What Computers Do

High-level Computer Languages and Compilers

Language Standards

Using C: Seven Steps

Programming Mechanics

## **2. Introducing C**

A Simple Example of C

The Example Explained

The Structure of a Simple Program

Tips on Making Your Programs Readable

Taking Another Step in Using C

While You're at It—Multiple Functions

Introducing Debugging

Keywords and Reserved Identifiers

Key Concepts

Programming Exercises

## **3. Data and C**

A Sample Program

Data Variables and Constants

Data: Data-Type Keywords

Basic C Data Types

Using Data Types

Arguments and Pitfalls

One More Example: Escape Sequences

Key Concepts

Programming Exercises

## **4. Character Strings and Formatted Input/Output**

Introductory Program

Character Strings: An Introduction

Constants and the C Preprocessor

Exploring and Exploiting printf() and scanf()

Key Concepts

Programming Exercises

## **5. Operators, Expressions, and Statements**

Introducing Loops

Fundamental Operators

Some Additional Operators

Expressions and Statements

Type Conversions

Function with Arguments

A Sample Program

Key Concepts  
Programming Exercises

## **6. C Control Statements: Looping**

Revisiting the while Loop  
The while Statement  
Which Is Bigger: Using Relational Operators and Expressions  
Indefinite Loops and Counting Loops  
The for Loop  
More Assignment Operators: +=, -=, \*=, /=, %=  
The Comma Operator  
An Exit-Condition Loop: do while  
Which Loop?  
Nested Loops  
Introducing Arrays  
A Loop Example Using a Function Return Value  
Key Concepts  
Programming Exercises

## **7. C Control Statements: Branching and Jumps**

The if Statement  
Adding else to the if Statement  
Let's Get Logical  
A Word-Count Program  
The Conditional Operator: ?:  
Loop Aids: continue and break  
Multiple Choice: switch and break  
The goto Statement  
Key Concepts  
Programming Exercises

## **8. Character Input/Output and Input Validation**

Single-Character I/O: getchar() and putchar()  
Buffers  
Terminating Keyboard Input  
Redirection and Files  
Creating a Friendlier User Interface  
Input Validation  
Menu Browsing  
Key Concepts  
Programming Exercises

## **9. Functions**

Reviewing Functions  
ANSI C Function Prototyping  
Recursion  
Compiling Programs with Two or More Source Code Files  
Finding Addresses: The & Operator  
Altering Variables in the Calling Function  
Pointers: A First Look  
Key Concepts

## Programming Exercises

### **10. Arrays and Pointers**

Arrays

Multidimensional Arrays

Pointers and Arrays

Functions, Arrays, and Pointers

Pointer Operations

Protecting Array Contents

Pointers and Multidimensional Arrays

Variable-Length Arrays (VLAs)

Compound Literals

Key Concepts

Programming Exercises

### **11. Character Strings and String Functions**

Representing Strings and String I/O

String Input

String Output

The Do-It-Yourself Option

String Functions

A String Example: Sorting Strings

The ctype.h Character Functions and Strings

Command-Line Arguments

String-to-Number Conversions

Key Concepts

Programming Exercises

### **12. Storage Classes, Linkage, and Memory Management**

Storage Classes

A Random-Number Function and a Static Variable

Roll 'Em

Allocated Memory: malloc() and free()

ANSI C Type Qualifiers

Key Concepts

Programming Exercises

### **13. File Input/Output**

Communicating with Files

Standard I/O

A Simple-Minded File-Condensing Program

File I/O: fprintf(), fscanf(), fgets(), and fputs()

Adventures in Random Access: fseek() and ftell()

Behind the Scenes with Standard I/O

Other Standard I/O Functions

Key Concepts

Programming Exercises