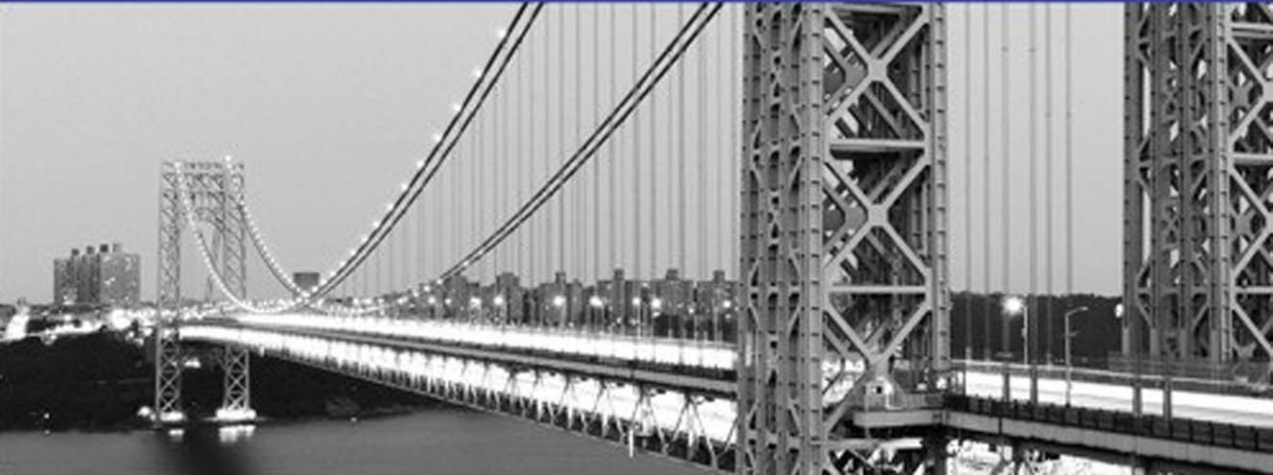
Stephen G. Kochan

Fourth Edition

Programming in C



Programming in C

Fourth Edition

Programming in C

Table of Contents

Table of Contents

Introduction

1 Some Fundamentals

- Programming
- Higher-Level Languages
- Operating Systems
- Compiling Programs
- Integrated Development Environments
- Language Interpreters

2 Compiling and Running Your First Program

- Compiling Your Program
- Running Your Program
- Understanding Your First Program
- Displaying the Values of Variables
- Comments
- Exercises

3 Variables, Data Types, and Arithmetic Expressions

- Understanding Data Types and Constants
  - The Integer Type int
  - The Floating Number Type float
  - The Extended Precision Type double

Table of Contents

The Single Character Type char

The Boolean Data Type _Bool

Type Specifiers: long, long long, short, unsigned, and signed

Working with Variables

Working with Arithmetic Expressions

Integer Arithmetic and the Unary Minus Operator

Combining Operations with Assignment: The Assignment Operators

Types _Complex and _Imaginary

Exercises

4 Program Looping

Triangular Numbers

The for Statement

Relational Operators

Aligning Output

Program Input

Nested for Loops

for Loop Variants

The while Statement

The do Statement

The break Statement

The continue Statement

Exercises

5 Making Decisions

The if Statement

The if-else Construct

Table of Contents

Compound Relational Tests

Nested if Statements

The else if Construct

The switch Statement

Boolean Variables

The Conditional Operator

Exercises

6 Working with Arrays

Defining an Array

Using Array Elements as Counters

Generating Fibonacci Numbers

Using an Array to Generate Prime Numbers

Initializing Arrays

Character Arrays

Base Conversion Using Arrays

The const Qualifier

Multidimensional Arrays

Variable Length Arrays

Exercises

7 Working with Functions

Defining a Function

Arguments and Local Variables

Function Prototype Declaration

Automatic Local Variables

Returning Function Results

Functions Calling Functions Calling...

Table of Contents

        - Declaring Return Types and Argument Types
        - Checking Function Arguments
    - Top-Down Programming
    - Functions and Arrays
        - Assignment Operators
        - Sorting Arrays
        - Multidimensional Arrays
    - Global Variables
    - Automatic and Static Variables
    - Recursive Functions
    - Exercises
- 8 Working with Structures
    - The Basics of Structures
    - A Structure for Storing the Date
        - Using Structures in Expressions
    - Functions and Structures
        - A Structure for Storing the Time
    - Initializing Structures
        - Compound Literals
    - Arrays of Structures
    - Structures Containing Structures
    - Structures Containing Arrays
    - Structure Variants
    - Exercises
- 9 Character Strings
    - Revisiting the Basics of Strings

Table of Contents

Arrays of Characters

Variable-Length Character Strings

- Initializing and Displaying Character Strings
- Testing Two Character Strings for Equality
- Inputting Character Strings
- Single-Character Input
- The Null String

Escape Characters

More on Constant Strings

Character Strings, Structures, and Arrays

- A Better Search Method

Character Operations

Exercises

10 Pointers

Pointers and Indirection

Defining a Pointer Variable

Using Pointers in Expressions

Working with Pointers and Structures

- Structures Containing Pointers
- Linked Lists

The Keyword const and Pointers

Pointers and Functions

Pointers and Arrays

- A Slight Digression About Program Optimization
- Is It an Array or Is It a Pointer?
- Pointers to Character Strings
- Constant Character Strings and Pointers

Table of Contents

The Increment and Decrement Operators Revisited

Operations on Pointers

Pointers to Functions

Pointers and Memory Addresses

Exercises

11 Operations on Bits

The Basics of Bits

Bit Operators

The Bitwise AND Operator

The Bitwise Inclusive-OR Operator

The Bitwise Exclusive-OR Operator

The Ones Complement Operator

The Left Shift Operator

The Right Shift Operator

A Shift Function

Rotating Bits

Bit Fields

Exercises

12 The Preprocessor

The #define Statement

Program Extendability

Program Portability

More Advanced Types of Definitions

The # Operator

The ## Operator

The #include Statement

System Include Files

Table of Contents

Conditional Compilation

The #ifdef, #endif, #else, and #ifndef Statements

The #if and #elif Preprocessor Statements

The #undef Statement

Exercises

13 Extending Data Types with the Enumerated Data Type, Type Definitions, and Data Type Conversions

Enumerated Data Types

The typedef Statement

Data Type Conversions

Sign Extension

Argument Conversion

Exercises

14 Working with Larger Programs

Dividing Your Program into Multiple Files

Compiling Multiple Source Files from the Command Line

Communication Between Modules

External Variables

Static Versus Extern Variables and Functions

Using Header Files Effectively

Other Utilities for Working with Larger Programs

The make Utility

The cvs Utility

Unix Utilities: ar, grep, sed, and so on

15 Input and Output Operations in C

Character I/O: getchar() and putchar()

Table of Contents

Formatted I/O: printf() and scanf()

The printf() Function

The scanf() Function

Input and Output Operations with Files

Redirecting I/O to a File

End of File

Special Functions for Working with Files

The fopen Function

The getc() and putc() Functions

The fclose() Function

The feof Function

The fprintf() and fscanf() Functions

The fgets() and fputs() Functions

stdin, stdout, and stderr

The exit() Function

Renaming and Removing Files

Exercises

16 Miscellaneous and Advanced Features

Miscellaneous Language Statements

The goto Statement

The null Statement

Working with Unions

The Comma Operator

Type Qualifiers

The register Qualifier

The volatile Qualifier

The restrict Qualifier

Table of Contents

Command-line Arguments

Dynamic Memory Allocation

The calloc() and malloc() Functions

The sizeof Operator

The free Function

Exercises

17 Debugging Programs

Debugging with the Preprocessor

Debugging Programs with gdb

Working with Variables

Source File Display

Controlling Program Execution

Getting a Stack Trace

Calling Functions and Setting Arrays and Structures

Getting Help with gdb Commands

Odds and Ends

18 Object-Oriented Programming

What Is an Object Anyway?

Instances and Methods

Writing a C Program to Work with Fractions

Defining an Objective-C Class to Work with Fractions

Defining a C++ Class to Work with Fractions

Defining a C# Class to Work with Fractions

A: C Language Summary

1.0 Digraphs and Identifiers

2.0 Comments

Table of Contents

- 3.0 Constants
- 4.0 Data Types and Declarations
- 5.0 Expressions
- 6.0 Storage Classes and Scope
- 7.0 Functions
- 8.0 Statements
- 9.0 The Preprocessor

B: The Standard C Library

- Standard Header Files
- String Functions
- Memory Functions
- Character Functions
- I/O Functions
- In-Memory Format Conversion Functions
- String-to-Number Conversion
- Dynamic Memory Allocation Functions
- Math Functions
- General Utility Functions

C: Compiling Programs with gcc

- General Command Format
- Command-Line Options

D: Common Programming Mistakes

E: Resources

- The C Programming Language
- C Compilers and Integrated Development Environments

Table of Contents

Miscellaneous
Index