

19.5	Design Issues for Abstract Data Types	502
	Naming Conventions	502
	Error Handling	502
	Generic ADTs	503
	ADTs in Newer Languages	503
20	LOW-LEVEL PROGRAMMING	509
20.1	Bitwise Operators	509
	Bitwise Shift Operators	510
	Bitwise Complement, <i>And</i> , Exclusive <i>Or</i> , and Inclusive <i>Or</i>	511
	Using the Bitwise Operators to Access Bits	512
	Using the Bitwise Operators to Access Bit-Fields	513
	Program: XOR Encryption	514
20.2	Bit-Fields in Structures	516
	How Bit-Fields Are Stored	517
20.3	Other Low-Level Techniques	518
	Defining Machine-Dependent Types	518
	Using Unions to Provide Multiple Views of Data	519
	Using Pointers as Addresses	520
	Program: Viewing Memory Locations	521
	The <code>volatile</code> Type Qualifier	523
21	THE STANDARD LIBRARY	529
21.1	Using the Library	529
	Restrictions on Names Used in the Library	530
	Functions Hidden by Macros	531
21.2	C89 Library Overview	531
21.3	C99 Library Changes	534
21.4	The <code><stddef.h></code> Header: Common Definitions	535
21.5	The <code><stdbool.h></code> Header (C99): Boolean Type and Values	536
22	INPUT/OUTPUT	539
22.1	Streams	540
	File Pointers	540
	Standard Streams and Redirection	540
	Text Files versus Binary Files	541
22.2	File Operations	543
	Opening a File	543
	Modes	544
	Closing a File	545
	Attaching a File to an Open Stream	546
	Obtaining File Names from the Command Line	546
	Program: Checking Whether a File Can Be Opened	547