

---

# Contenido

---

<b>Prefacio</b>		<b>xxxi</b>
<b>Capítulo 1</b>	<b>Conceptos de computación</b>	<b>1</b>
1.1	Introducción	2
1.2	¿Qué es una computadora?	4
1.3	Organización de la computadora	4
1.4	Procesamiento por lotes, programación múltiple y tiempo compartido	5
1.5	Computación personal, computación distribuida y computación cliente/servidor	6
1.6	Lenguajes máquina, lenguajes ensambladores y lenguajes de alto nivel	6
1.7	La historia de C	7
1.8	La biblioteca estándar de C	8
1.9	Otros lenguajes de alto nivel	9
1.10	Programación estructurada	9
1.11	Los fundamentos del entorno de C	10
1.12	Notas generales sobre C y este libro	10
1.13	C concurrente	12
1.14	Programación orientada a objetos y C++	14
	<i>Resumen • Terminología • Prácticas sanas de programación • Sugerencias de portabilidad • Sugerencias de rendimiento • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios • Lecturas recomendadas</i>	
<b>Capítulo 2</b>	<b>Introducción a la programación en C</b>	<b>23</b>
2.1	Introducción	24
2.2	Un programa simple en C: imprimir una línea de texto	24
2.3	Otro programa simple en C: sumar dos enteros	28
2.4	Conceptos de memoria	33
2.5	Aritmética en C	34
2.6	Toma de decisiones: operadores de igualdad y relacionales	37

	<i>Resumen • Terminología • Errores comunes de programación • Prácticas sanas de programación • Sugerencia de portabilidad • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
<b>Capítulo 3</b>	<b>Desarrollo de programas estructurados</b>	<b>55</b>
3.1	Introducción	56
3.2	Algoritmos	56
3.3	Pseudocódigo	57
3.4	Estructuras de control	58
3.5	La estructura de selección If	60
3.6	La estructura de selección If/Else	61
3.7	La estructura de repetición While	65
3.8	Cómo formular algoritmos: Estudio de caso 1 (repetición controlada por contador)	67
3.9	Cómo formular algoritmos con refinamiento descendente paso a paso: Estudio de caso 2 (repetición controlada por centinela)	69
3.10	Cómo formular algoritmos con refinamiento descendente paso a paso: Estudio de caso 3 (estructuras de control anidadas)	74
3.11	Operadores de asignación	77
3.12	Operadores incrementales y decrementales	79
	<i>Resumen • Terminología • Errores comunes de programación • Prácticas sanas de programación • Sugerencias de rendimiento • Observaciones de ingeniería de software • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
<b>Capítulo 4</b>	<b>Control de programa</b>	<b>101</b>
4.1	Introducción	102
4.2	Lo esencial de la repetición	102
4.3	Repetición controlada por contador	103
4.4	La estructura de repetición for	105
4.5	La estructura for: Notas y observaciones	108
4.6	Ejemplos utilizando la estructura for	108
4.7	La estructura de selección múltiple Switch	112
4.8	La estructura de repetición do/while	118
4.9	Los enunciados break y continue	120
4.10	Operadores lógicos	122
4.11	Confusión entre los operadores de igualdad (==) y de asignación (=)	124
4.12	Resumen de programación estructurada	126
	<i>Resumen • Terminología • Errores comunes de programación • Prácticas sanas de programación • Sugerencias de rendimiento • Sugerencias de portabilidad • Observaciones de ingeniería de software • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	

<b>Capítulo 5</b>	<b>Funciones</b>	<b>147</b>
5.1	Introducción	148
5.2	Módulos de programa en C	148
5.3	Funciones matemáticas de biblioteca	149
5.4	Funciones	150
5.5	Definiciones de función	152
5.6	Prototipo de funciones	155
5.7	Archivos de cabecera	159
5.8	Cómo llamar funciones: llamada por valor y llamada por referencia	160
5.9	Generación de números aleatorios	160
5.10	Ejemplo: un juego de azar	165
5.11	Clases de almacenamiento	168
5.12	Reglas de alcance	170
5.13	Recursión	171
5.14	Ejemplo utilizando recursión: la serie Fibonacci	176
5.15	Recursión en comparación con iteración	180
	<i>Resumen • Terminología • Errores comunes de programación • Prácticas sanas de programación • Sugerencias de portabilidad • Sugerencias de rendimiento • Observaciones de ingeniería de software • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
<b>Capítulo 6</b>	<b>Arreglos</b>	<b>203</b>
6.1	Introducción	204
6.2	Arreglos	204
6.3	Cómo declarar arreglos	206
6.4	Ejemplos utilizando arreglos	207
6.5	Cómo pasar arreglos a funciones	217
6.6	Cómo ordenar arreglos	223
6.7	Estudio de caso: Cómo calcular el promedio, la mediana y el modo utilizando arreglos	225
6.8	Búsqueda en arreglos	228
6.9	Arreglos con múltiples subíndices	231
	<i>Resumen • Terminología • Errores comunes de programación • Prácticas sanas de programación • Sugerencias de rendimiento • Observaciones de ingeniería de software • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios • Ejercicios de recursión</i>	
<b>Capítulo 7</b>	<b>Punteros</b>	<b>259</b>
7.1	Introducción	260
7.2	Declaraciones e inicialización de variables de apuntadores	260
7.3	Operadores de apuntador	261
7.4	Cómo llamar funciones por referencia	263
7.5	Cómo usar el calificador const con apuntadores	268

7.6	Ordenamiento de tipo burbuja utilizando llamadas por referencia	272
7.7	Expresiones de punteros y aritmética de apuntadores	277
7.8	Relación entre apuntadores y arreglos	281
7.9	Arreglos de apuntadores	284
7.10	Estudio de caso: simulación de barajar y repartir cartas	286
7.11	apuntadores a funciones	291
	<i>Resumen • Terminología • Errores comunes de programación • Prácticas sanas de programación • Sugerencias de rendimiento • Sugerencias de portabilidad • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios • Sección especial: cómo construir su propia computadora</i>	
<b>Capítulo 8</b>	<b>Caracteres y cadenas</b>	<b>317</b>
8.1	Introducción	318
8.2	Fundamentos de cadenas y caracteres	318
8.3	Biblioteca de manejo de caracteres	320
8.4	Funciones de conversión de cadenas	325
8.5	Funciones de la biblioteca estándar de entradas/salidas	330
8.6	Funciones de manipulación de cadenas de la biblioteca de manejo de cadenas	333
8.7	Funciones de comparación de la biblioteca de manejo de cadenas	336
8.8	Funciones de búsqueda de la biblioteca de manejo de cadenas	338
8.9	Funciones de memoria de la biblioteca de manejo de cadenas	344
8.10	Otras funciones de la biblioteca de manejo de cadenas	347
	<i>Resumen • Terminología • Errores comunes de programación • Prácticas sanas de programación • Sugerencias de portabilidad • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios • Sección especial: Un compendio de ejercicios más avanzados de manipulación de cadenas</i>	
<b>Capítulo 9</b>	<b>Entrada/Salida con formato</b>	<b>365</b>
9.1	Introducción	366
9.2	Flujos	366
9.3	Salida con formato mediante printf	367
9.4	Cómo imprimir enteros	367
9.5	Cómo imprimir números de punto flotante	369
9.6	Cómo imprimir cadenas y caracteres	371
9.7	Otros especificadores de conversión	372
9.8	Cómo imprimir con anchos de campo y precisiones	372
9.9	Uso de banderas en la cadena de control de formato printf	375
9.10	Cómo imprimir literales y secuencias de escape	377
9.11	Formato de entrada con scanf	379
	<i>Resumen • Terminología • Errores comunes de programación • Prácticas sanas de programación • Sugerencias de portabilidad • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	

<b>Capítulo 10</b>	<b>Estructuras, uniones, manipulaciones de bits y enumeraciones</b>	<b>395</b>
10.1	Introducción	396
10.2	Definiciones de estructura	396
10.3	Cómo inicializar estructuras	399
10.4	Cómo tener acceso a miembros de estructuras	399
10.5	Cómo utilizar estructuras con funciones	401
10.6	Typedef	401
10.7	Ejemplo: Simulación de barajar y distribuir cartas de alto rendimiento	402
10.8	Uniones	402
10.9	Operadores a nivel de bits	406
10.10	Campos de bits	414
10.11	Constantes de enumeración	416
	<i>Resumen • Terminología • Errores comunes de programación • Prácticas sanas de programación • Sugerencias de portabilidad • Sugerencias de rendimiento • Observación sobre ingeniería de software • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
<b>Capítulo 11</b>	<b>Procesamiento de archivos</b>	<b>431</b>
11.1	Introducción	432
11.2	La jerarquía de datos	432
11.3	Archivos y flujos	434
11.4	Cómo crear un archivo de acceso secuencial	435
11.5	Cómo leer datos de un archivo de acceso secuencial	440
11.6	Archivos de acceso directo	445
11.7	Cómo crear un archivo de acceso directo	446
11.8	Cómo escribir datos directamente a un archivo de acceso directo	448
11.9	Cómo leer datos directamente de un archivo de acceso directo	450
11.10	Estudio de caso: Un programa de procesamiento de transacciones	451
	<i>Resumen • Terminología • Errores comunes de programación • Prácticas sanas de programación • Sugerencias de rendimiento • Sugerencia de portabilidad • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
<b>Capítulo 12</b>	<b>Estructuras de datos</b>	<b>467</b>
12.1	Introducción	468
12.2	Estructuras autoreferenciadas	469
12.3	Asignación dinámica de memoria	470
12.4	Listas enlazadas	471
12.5	Pilas	479
12.6	Colas de espera	484
12.7	Arboles	489
	<i>Resumen • Terminología • Errores comunes de programación • Prácticas sanas de programación • Sugerencias de rendimiento •</i>	

	<i>Sugerencia de portabilidad • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
<b>Capítulo 13</b>	<b>El preprocesador</b>	<b>521</b>
13.1	Introducción	522
13.2	La directiva de preprocesador #include	522
13.3	La directiva de preprocesador #define: constantes simbólicas	523
13.4	La directiva de preprocesador #define: macros	523
13.5	Compilación condicional	525
13.6	Las directivas de preprocesador #error y #pragma	526
13.7	Los operadores # y ##	527
13.8	Números de línea	527
13.9	Constantes simbólicas predefinidas	528
13.10	Asertos	528
	<i>Resumen • Terminología • Errores comunes de programación • Prácticas sanas de programación • Sugerencia de rendimiento • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
<b>Capítulo 14</b>	<b>Temas avanzados</b>	<b>535</b>
14.1	Introducción	536
14.2	Cómo redirigir entradas/salidas en sistemas UNIX y DOS	536
14.3	Listas de argumentos de longitud variable	537
14.4	Cómo utilizar argumentos en línea de comandos	540
14.5	Notas sobre la compilación de programas con varios archivos fuente	540
14.6	Terminación de programas mediante Exit y Atexit	543
14.7	El calificador de tipo volátil	543
14.8	Sufijos para constantes de enteras y punto flotante	543
14.9	Más sobre archivos	545
14.10	Manejo de señales	547
14.11	Asignación dinámica de memoria: funciones calloc y realloc	548
14.12	La bifurcación incondicional: Goto	548
	<i>Resumen • Terminología • Error común de programación • Sugerencias de portabilidad • Sugerencias de rendimiento • Observaciones de ingeniería de software • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
<b>Capítulo 15</b>	<b>C++ como un “C mejorado”</b>	<b>559</b>
15.1	Introducción	560
15.2	Comentarios de una sola línea de C++	561
15.3	Flujo de entrada/salida de C++	562
15.4	Declaraciones en C++	563
15.5	Cómo crear nuevos tipos de datos en C++	564
15.6	Prototipos de funciones y verificación de tipo	565

15.7	Funciones en línea	566
15.8	Parámetros de referencia	569
15.9	El calificador Const	574
15.10	Asignación dinámica de memoria mediante new y delete	576
15.11	Argumentos por omisión	578
15.12	Operador de resolución de alcance unario	578
15.13	Homonimia de funciones	579
15.14	Especificaciones de enlace	582
15.15	Plantillas de función	583
	<i>Resumen • Terminología • Errores comunes de programación • Prácticas sanas de programación • Sugerencia de rendimiento • Sugerencias de portabilidad • Observación de ingeniería de software • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
<b>Capítulo 16</b>	<b>Clases y abstracción de datos</b>	<b>593</b>
16.1	Introducción	594
16.2	Definiciones de estructuras	596
16.3	Cómo tener acceso a miembros de estructuras	597
16.4	Cómo poner en práctica mediante un struct un tipo Time definido por el usuario	597
16.5	Cómo implantar un tipo de dato abstracto Time con una clase	599
16.6	Alcance de clase y acceso a miembros de clase	605
16.7	Cómo separar el interfaz de una puesta en práctica	606
16.8	Cómo controlar el acceso a miembros	608
16.9	Funciones de acceso y funciones de utilería	613
16.10	Cómo inicializar objetos de clase: constructores	614
16.11	Cómo utilizar argumentos por omisión con los constructores	616
16.12	Cómo utilizar destructores	617
16.13	Cuándo son llamados los destructores y los constructores	619
16.14	Cómo utilizar miembros de datos y funciones miembro	621
16.15	Una trampa sutil: cómo regresar una referencia a un miembro de datos privado	626
16.16	Asignación por omisión en copia a nivel de miembro	629
16.17	Reutilización del software	631
	<i>Resumen • Terminología • Errores comunes de programación • Prácticas sanas de programación • Sugerencias de rendimiento • Observaciones sobre ingeniería de software • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
<b>Capítulo 17</b>	<b>Clases: Parte II</b>	<b>641</b>
17.1	Introducción	642
17.2	Objetos constantes y funciones de miembro const	642

17.3	Composición: clases como miembros de otras clases	648
17.4	Funciones amigo y clases amigo	650
17.5	Cómo utilizar el apuntador this	655
17.6	Asignación dinámica de memoria mediante los operadores new y delete	660
17.7	Miembros de clase estáticos	661
17.8	Abstracción de datos y ocultamiento de información	665
17.8.1	Ejemplo: Tipo de datos abstracto de arreglo	666
17.8.2	Ejemplo: Tipo de datos abstracto de cadena	667
17.8.3	Ejemplo: Tipo de datos abstracto de cola	667
17.9	Clases contenedor e iteradores	668
17.10	Clases plantilla	668
	<i>Resumen • Terminología • Errores comunes de programación • Prácticas sanas de programación • Sugerencias de rendimiento • Sugerencia de portabilidad • Observaciones sobre ingeniería de software • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
<b>Capítulo 18</b>	<b>Homonimia de operadores</b>	<b>679</b>
18.1	Introducción	680
18.2	Fundamentos de la homonimia de operadores	681
18.3	Restricciones sobre la homonimia de operadores	682
18.4	Funciones operador como miembros de clase en comparación con funciones amigo	684
18.5	Cómo hacer la homonimia de operadores de inserción de flujo y de extracción de flujo	685
18.6	Homonimia de operadores unarios	687
18.7	Homonimia de operadores binarios	688
18.8	Estudio de caso: una clase de Array	689
18.9	Conversión entre tipos	698
18.10	Estudio de caso: una clase de String	700
18.11	Homonimia de ++ y --	709
18.12	Estudio de caso: una clase de Date	712
	<i>Resumen Terminología • Errores comunes de programación • Prácticas sanas de programación • Sugerencias de rendimiento • Observaciones sobre ingeniería de software • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
<b>Capítulo 19</b>	<b>Herencia</b>	<b>729</b>
19.1	Introducción	730
19.2	Clases base y clases derivadas	732
19.3	Miembros protegidos	734
19.4	Cómo hacer la conversión explícita (cast) de apuntadores de clase base a apuntadores de clase derivada	734
19.5	Cómo utilizar funciones miembro	738

19.6	Cómo redefinir los miembros de clase base en una clase derivada	739
19.7	Clases base públicas, protegidas y privadas	743
19.8	Clases base directas y clases base indirectas	743
19.9	Cómo utilizar constructores y destructores en clases derivadas	743
19.10	Conversión implícita de objeto de clase derivada a objeto de clase base	745
19.11	Ingeniería de software con herencia	748
19.12	Composición en comparación con herencia	749
19.13	Relaciones “utiliza un” y “conoce un”	750
19.14	Estudio de caso: Point, Circle, cylinder	750
19.15	Herencia múltiple	755
	<i>Resumen • Terminología • Errores comunes de programación • Prácticas sanas de programación • Sugerencia de rendimiento • Observaciones de ingeniería de software • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
<b>Capítulo 20</b>	<b>Funciones virtuales y polimorfismo</b>	<b>769</b>
20.1	Introducción	770
20.2	Campos de tipo y enunciados switch	770
20.3	Funciones virtuales	771
20.4	Clases base abstractas y clases concretas	772
20.5	Polimorfismo	773
20.6	Estudio de caso: un sistema de nómina utilizando polimorfismo	774
20.7	Clases nuevas y ligadura dinámica	781
20.8	Destructores virtuales	785
20.9	Estudio de caso: cómo heredar interfaz, y puesta en práctica	785
	<i>Resumen • Terminología • Errores comunes de programación • Prácticas sanas de programación • Sugerencias de rendimiento • Observaciones de ingeniería de software • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
<b>Capítulo 21</b>	<b>Flujo de entrada/salida de C++</b>	<b>797</b>
21.1	Introducción	799
21.2	Flujos	799
	21.2.1 Archivos de cabecera de biblioteca iostream	800
	21.2.2 Clases y objetos de flujo de entrada/salida	800
21.3	Salida de flujo	802
	21.3.1 Operador de inserción de flujo	802
	21.3.2 Cómo concatenar operadores de inserción/extracción de flujo	804
	21.3.3 Salida de variables char*	805
	21.3.4 Extracción de caracteres mediante la función de miembro put; cómo concatenar put	805
21.4	Entrada de flujo	806

	21.4.1	Operador de extracción de flujo	806
	21.4.2	Funciones de miembro get y getline	809
	21.4.3	Otras funciones miembro istream (peek, putback, ignore)	811
	21.4.4	Entrada/Salida de tipo seguro	812
21.5		Entrada/Salida sin formato mediante read, gcount y write	812
21.6		Manipuladores de flujo	812
	21.6.1	Base de flujo integral: manipuladores de flujo dec, oct, hex y setbase	812
	21.6.2	Precisión de punto flotante (precision, setprecision)	813
	21.6.3	Ancho de campo (setw, width)	814
	21.6.4	Manipuladores definidos por el usuario	816
21.7		Estados de formato de flujo	816
	21.7.1	Banderas de estado de formato (setf, unsetf, flags)	818
	21.7.2	Ceros a la derecha y puntos decimales (ios::showpoint)	818
	21.7.3	Justificación (ios::left, ios::right, ios::internal)	819
	21.7.4	Relleno (fill, setfill)	821
	21.7.5	Base de flujo integral (ios::dec, ios::oct, ios::hex, ios::showbase)	821
	21.7.6	Números de punto flotante; notación científica (ios::scientific, ios::fixed)	823
	21.7.7	Control mayúsculas/minúsculas (ios::uppercase)	824
	21.7.8	Cómo activar y desactivar las banderas de formato (flags, setiosflags, resetiosflags)	824
21.8		Estados de errores de flujo	825
21.9		Entradas/salidas de tipos definidos por usuario	827
21.10		Cómo ligar un flujo de salida con un flujo de entrada	829
		<i>Resumen Terminología • Errores comunes de programación • Prácticas sanas de programación • Consejos para mejorar el rendimiento • Sugerencias de portabilidad • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
<b>Apéndice A</b>		<b>Sintaxis de C</b>	<b>846</b>
	A.1	Gramática lexicográfica	846
	A.2	Gramática de estructural de frases	850
	A.3	Directrices de preprocesador	856
<b>Apéndice B</b>		<b>Biblioteca estándar</b>	<b>858</b>
	B.1	Errores <errno.h>	858
	B.2	Definiciones comunes <stddef.h>	858
	B.3	Diagnósticos <assert.h>	859
	B.4	Manejo de caracteres <ctype.h>	859
	B.5	Localización <locale.h>	860
	B.6	Matemáticas <math.h>	863
	B.7	Saltos no locales <setjmp.h>	865

B.8	Manejo de señales <signal.h>	865
B.9	Argumentos variables <stdarg.h>	867
B.10	Entrada/salida <stdio.h>	867
B.11	Utilerías generales <stdlib.h>	875
B.12	Manejo de cadenas <string.h>	881
B.13	Fecha y hora <time.h>	884
B.14	Límites de puesta en práctica:	
	<limits.h>	887
	<float.h>	887
<b>Apéndice C</b>	<b>Precedencia y asociatividad de operadores</b>	<b>890</b>
<b>Apéndice D</b>	<b>Conjunto de caracteres ASCII</b>	<b>891</b>
<b>Apéndice E</b>	<b>Sistemas numéricos</b>	<b>893</b>
E.1	Introducción	894
E.2	Cómo abreviar números binarios como octales y hexadecimales	897
E.3	Cómo convertir números octales y hexadecimales a binarios	898
E.4	Cómo convertir de binario, octal y hexadecimal a decimal	898
E.5	Cómo convertir de decimal a binario, octal o hexadecimal	899
E.6	Números binarios negativos: notación complementaria a dos	901
	<i>Resumen • Terminología • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
<b>Índice</b>		<b>909</b>