
Contenido

Prefacio

xxxii

Capítulo 1	Introducción a las computadoras y a la programación en C++	1
1.1	Introducción	3
1.2	¿Qué es una computadora?	5
1.3	Organización de las computadoras	5
1.4	Evolución de los sistemas operativos	6
1.5	Computación personal, distribuida y cliente/servidor	7
1.6	Lenguajes de máquina, lenguajes ensambladores y lenguajes de alto nivel	8
1.7	Historia de C y C++	9
1.8	Biblioteca estándar de C++	10
1.9	Java, Internet y World Wide Web	11
1.10	Otros lenguajes de alto nivel	12
1.11	Programación estructurada	12
1.12	Fundamentos del entorno C++ típico	13
1.13	Apuntes generales sobre C++ y esta obra	15
1.14	Introducción a la programación en C++	16
1.15	Un programa sencillo: impresión de una línea de texto	17
1.16	Otro programa sencillo: suma de dos enteros	21
1.17	Conceptos sobre memoria	25
1.18	Aritmética	26
1.19	Toma de decisiones: operadores de igualdad y relacionales	30
1.20	Archivos de encabezado de nuevo estilo y espacios de nombres	34
1.21	Pensando en objetos	35
	<i>Resumen • Terminología • Errores comunes de programación • Buenas prácticas de programación • Tip de desempeño • Tips de portabilidad • Observaciones de ingeniería de software • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	

Capítulo 2	Estructuras de control	54
2.1	Introducción	56
2.2	Algoritmos	57
2.3	Pseudocódigo	57
2.4	Estructuras de control	58
2.5	La estructura de selección if	60
2.6	La estructura de selección if/else	62
2.7	La estructura de repetición while	66
2.8	Formulación de algoritmos: caso de estudio 1 (repetición controlada por contador)	68
2.9	Formulación de algoritmos con refinación descendente paso a paso: caso de estudio 2 (repetición controlada por centinela)	70
2.10	Formulación de algoritmos con refinación descendente paso a paso: caso de estudio 3 (estructuras de control anidadas)	77
2.11	Operadores de asignación	81
2.12	Operadores de incremento y decremento	82
2.13	Fundamentos de la repetición controlada por contador	85
2.14	La estructura de repetición for	87
2.15	Ejemplos del uso de la estructura for	91
2.16	La estructura switch de selección múltiple	95
2.17	La estructura de repetición do/while	102
2.18	Las instrucciones break y continue	104
2.19	Operadores lógicos	105
2.20	Confusión de los operadores de igualdad (==) y de asignación (=)	108
2.21	Resumen de la programación estructurada	110
2.22	Pensando en objetos: identificación de los objetos de un problema <i>Resumen • Terminología • Errores comunes de programación • Buenas prácticas de programación • Tips de desempeño • Tips de portabilidad • Observaciones de ingeniería de software • Tips de prueba y depuración • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	116
Capítulo 3	Funciones	142
3.1	Introducción	144
3.2	Componentes de los programas en C++	145
3.3	Funciones de la biblioteca matemática	145
3.4	Funciones	147
3.5	Definiciones de funciones	148
3.6	Prototipos de función	152
3.7	Archivos de encabezado	155
3.8	Generación de números aleatorios	157
3.9	Ejemplo: juego de azar y presentación de enum	162
3.10	Clases de almacenamiento	166
3.11	Reglas de alcance	168
3.12	Recursión	172
3.13	Ejemplo con recursión: serie de Fibonacci	175

3.14	Recursión en comparación con iteración	178
3.15	Funciones con listas de parámetros vacías	181
3.16	Funciones en línea	182
3.17	Referencias y parámetros de referencia	183
3.18	Argumentos predeterminados	187
3.19	Operador unario de resolución de alcance	189
3.20	Sobrecarga de funciones	190
3.21	Plantillas de funciones	192
3.22	Pensando en objetos: identificación de los atributos de un objeto	194
	<i>Resumen • Terminología • Errores comunes de programación • Buenas prácticas de programación • Tip de desempeño • Tips de portabilidad • Observaciones de ingeniería de software • Tip de prueba y depuración • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
Capítulo 4	Arreglos	222
4.1	Introducción	224
4.2	Arreglos	224
4.3	Declaración de arreglos	226
4.4	Ejemplos que utilizan arreglos	227
4.5	Paso de arreglos a funciones	241
4.6	Ordenamiento de arreglos	245
4.7	Caso de estudio: cálculo de media, mediana y moda a través de arreglos	247
4.8	Búsqueda en arreglos: búsqueda lineal y búsqueda binaria	252
4.9	Arreglos de múltiples subíndices	256
4.10	Pensando en objetos: identificación de los comportamientos de una clase	263
	<i>Resumen • Terminología • Errores comunes de programación • Buenas prácticas de programación • Tips de desempeño • Tip de portabilidad • Observaciones de ingeniería de software • Tips de prueba y depuración • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios • Ejercicios de recursión</i>	
Capítulo 5	Apuntadores y cadenas	282
5.1	Introducción	284
5.2	Declaración e inicialización de variables de apuntador	285
5.3	Operadores de los apuntadores	286
5.4	Llamado de funciones por referencia	289
5.5	Empleo del calificador <code>const</code> con apuntadores	293
5.6	Ordenamiento de burbuja mediante llamada por referencia	299
5.7	Expresiones de apuntadores y aritmética de apuntadores	304
5.8	Relación entre apuntadores y arreglos	307
5.9	Arreglos de apuntadores	311
5.10	Caso de estudio: simulación de barajado y repartición de naipes	312
5.11	Apuntadores a funciones	317
5.12	Introducción al procesamiento de caracteres y cadenas	322
5.12.1	Fundamentos de los caracteres y las cadenas	322

5.12.2	Funciones de manipulación de cadenas de la biblioteca de manejo de cadenas	324
5.13	Pensando en objetos: interacciones entre los objetos	331
	<i>Resumen • Terminología • Errores comunes de programación • Buenas prácticas de programación • Tips de desempeño • Tips de portabilidad • Observaciones de ingeniería de software • Tip de prueba y depuración • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios • Sección especial: construya su propia computadora • Más ejercicios de apuntadores • Ejercicios de manipulación de cadenas • Sección especial: manipulación avanzada de cadenas • Interesante proyecto de manipulación de cadenas</i>	
Capítulo 6	Clases y abstracción de datos	362
6.1	Introducción	364
6.2	Definiciones de estructuras	365
6.3	Acceso a los miembros de las estructuras	366
6.4	Implementación del tipo <code>Time</code> definido por el usuario mediante una estructura	367
6.5	Implementación del tipo de datos abstracto <code>Time</code> mediante una clase	369
6.6	Alcance de clase y acceso a los miembros de una clase	376
6.7	Separación de interfaz e implementación	378
6.8	Control de acceso a los miembros	381
6.9	Funciones de acceso y de utilidad	384
6.10	Inicialización de objetos de clase: constructores	387
6.11	Uso de argumentos predeterminados con los constructores	388
6.12	Uso de los destructores	391
6.13	Invocación de constructores y destructores	392
6.14	Empleo de los datos miembro y de las funciones miembro	395
6.15	Trampa sutil: devolución de una referencia a un dato miembro <code>private</code>	400
6.16	Asignación por copia predeterminada a nivel de miembro	402
6.17	Reutilización de software	404
6.18	Pensando en objetos: programación de las clases para el simulador del elevador	404
	<i>Resumen • Terminología • Errores comunes de programación • Buenas prácticas de programación • Tips de desempeño • Observaciones de ingeniería de software • Tips de prueba y depuración • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
Capítulo 7	Clases: parte II	416
7.1	Introducción	418
7.2	Objetos <code>const</code> (constantes) y funciones miembro <code>const</code>	418
7.3	Composición: los objetos como miembros de clases	426
7.4	Funciones <code>friend</code> y clases <code>friend</code>	432
7.5	Uso del apunador <code>this</code>	435
7.6	Asignación dinámica de memoria con los operadores <code>new</code> y <code>delete</code>	440
7.7	Miembros de clase <code>static</code>	441
7.8	Abstracción de datos y ocultamiento de información	446

7.8.1	Ejemplo: tipo de datos abstracto de arreglo	448
7.8.2	Ejemplo: tipo de datos abstracto de cadena	449
7.8.3	Ejemplo: tipo de datos abstracto de cola	449
7.9	Clases contenedoras e iteradores	450
7.10	Clases proxy	450
7.11	Pensando en objetos: composición y administración dinámica de objetos en el simulador del elevador	452
	<i>Resumen • Terminología • Errores comunes de programación • Buenas prácticas de programación • Tips de desempeño • Observaciones de ingeniería de software • Tips de prueba y depuración • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
Capítulo 8	Sobrecarga de operadores	462
8.1	Introducción	464
8.2	Fundamentos de la sobrecarga de operadores	465
8.3	Restricciones de la sobrecarga de operadores	466
8.4	Comparación de las funciones de operador como miembros de clase y como funciones friend	468
8.5	Sobrecarga de operadores de inserción y extracción de flujo	469
8.6	Sobrecarga de operadores unarios	472
8.7	Sobrecarga de operadores binarios	473
8.8	Caso de estudio: clase Array	473
8.9	Conversión entre tipos	486
8.10	Caso de estudio: clase String	487
8.11	Sobrecarga de ++ y --	498
8.12	Caso de estudio: clase Date	499
	<i>Resumen • Terminología • Errores comunes de programación • Buenas prácticas de programación • Tips de desempeño • Observaciones de ingeniería de software • Tip de prueba y depuración • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
Capítulo 9	Herencia	518
9.1	Introducción ¹	520
9.2	Herencia: clases base y clases derivadas	522
9.3	Miembros protected	524
9.4	Conversión mediante cast de apuntadores a clases base a apuntadores a clases derivadas	524
9.5	Uso de funciones miembro	530
9.6	Sobreposición de los miembros de clase base en una clase derivada	530
9.7	Herencia public , protected y private	535
9.8	Clases base directas e indirectas	536
9.9	Uso de los constructores y destructores en las clases derivadas	536
9.10	Conversión implícita de objetos de clase derivada a objetos de clase base	540
9.11	Ingeniería de software con herencia	541

9.12	Composición en comparación con herencia	542
9.13	Relaciones “usa un” y “conoce un”	543
9.14	Caso de estudio: punto, círculo, cilindro	543
9.15	Herencia múltiple	550
	<i>Resumen • Terminología • Errores comunes de programación • Buena práctica de programación • Tips de desempeño • Observaciones de ingeniería de software • Ejercicio de autoevaluación • Respuesta al ejercicio de autoevaluación • Ejercicios</i>	
Capítulo 10	Funciones virtuales y polimorfismo	562
10.1	Introducción	564
10.2	Campos de tipo e instrucciones switch	564
10.3	Funciones virtual	565
10.4	Clases base abstractas y clases concretas	566
10.5	Polimorfismo	566
10.6	Caso de estudio: un sistema de nómina que utiliza el polimorfismo	569
10.7	Clases nuevas y enlace dinámico	580
10.8	Destructores virtual	580
10.9	Caso de estudio: herencia de la interfaz y de la implementación	581
10.10	Polimorfismo, funciones virtual y enlace dinámico “por debajo del agua”	589
	<i>Resumen • Terminología • Errores comunes de programación • Buenas prácticas de programación • Tips de desempeño • Observaciones de ingeniería de software • Ejercicio de autoevaluación • Respuesta al ejercicio de autoevaluación • Ejercicios</i>	
Capítulo 11	Entrada/salida de flujo en C++	598
11.1	Introducción	601
11.2	Flujos	601
11.2.1	Archivos de encabezado de la biblioteca iostream	602
11.2.2	Clases y objetos de entrada/salida de flujo	602
11.3	Salida de flujo	604
11.3.1	Operador de inserción de flujo	604
11.3.2	Puesta en cascada de los operadores de inserción/extracción de flujo	606
11.3.3	Salida de variables char *	607
11.3.4	Salida de caracteres con la función miembro put ; put en cascada	608
11.4	Entrada de flujo	608
11.4.1	Operador de extracción de flujo	609
11.4.2	Las funciones miembro get y getline	610
11.4.3	Las funciones miembro peek , putback e ignore de istream	614
11.4.4	E/S a prueba de tipos	614
11.5	E/S sin formato mediante read , gcount y write	614
11.6	Manipuladores de flujo	615
11.6.1	Base de flujo integral: dec , oct , hex y setbase	616
11.6.2	Precisión de punto flotante (precision , setprecision)	616
11.6.3	Anchura de campo (setw , width)	617
11.6.4	Manipuladores definidos por el usuario	618

11.7	Estados de formato de flujo	619
11.7.1	Indicadores de estado de formato	619
11.7.2	Ceros a la derecha y puntos decimales (<code>ios::showpoint</code>)	621
11.7.3	Alineación (<code>ios::left</code> , <code>ios::right</code> , <code>ios::internal</code>)	622
11.7.4	Relleno (<code>fill</code> , <code>setfill</code>)	624
11.7.5	Base de flujo integral (<code>ios::dec</code> , <code>ios::oct</code> , <code>ios::hex</code> , <code>ios::showbase</code>)	625
11.7.6	Números de punto flotante, notación científica (<code>ios::scientific</code> , <code>ios::fixed</code>)	626
11.7.7	Control de mayúsculas/minúsculas (<code>ios::uppercase</code>)	627
11.7.8	Establecimiento y restablecimiento de los indicadores de formato (<code>flags</code> , <code>setiosflags</code> , <code>resetiosflags</code>)	628
11.8	Estados de error de flujo	630
11.9	Enlace de un flujo de salida a un flujo de entrada	632
	<i>Resumen • Terminología • Errores comunes de programación • Buenas prácticas de programación • Tip de desempeño • Tip de portabilidad • Observaciones de ingeniería de software • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
Capítulo 12	Plantillas	646
12.1	Introducción	648
12.2	Plantillas de función	649
12.3	Sobrecarga de funciones de plantilla	652
12.4	Plantillas de clase	652
12.5	Plantillas de clase y parámetros sin tipo	658
12.6	Las plantillas y la herencia	659
12.7	Las plantillas y los <code>friend</code>	659
12.8	Las plantillas y los miembros <code>static</code>	661
	<i>Resumen • Terminología • Errores comunes de programación • Tips de desempeño • Observaciones de ingeniería de software • Tip de prueba y depuración • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
Capítulo 13	Manejo de excepciones	666
13.1	Introducción	668
13.2	Cuándo debe utilizarse el manejo de excepciones	670
13.3	Otras técnicas para el manejo de errores	671
13.4	Puntos básicos del manejo de excepciones en C++: <code>try</code> , <code>throw</code> y <code>catch</code>	672
13.5	Un ejemplo simple del manejo de excepciones: la división entre cero	673
13.6	Lanzamiento de una excepción	675
13.7	Cómo atrapar una excepción	676
13.8	Cómo volver a lanzar una excepción	679
13.9	Especificación de excepciones	681
13.10	Procesamiento de excepciones inesperadas	682
13.11	Vaciado de la pila	682

13.12	Constructores, destructores y manejo de excepciones	684
13.13	Las excepciones y la herencia	684
13.14	Procesamiento de las fallas de <code>new</code>	685
13.15	La clase <code>auto_ptr</code> y la asignación dinámica de memoria	688
13.16	La jerarquía de excepciones de la biblioteca estándar	690
	<i>Resumen • Terminología • Errores comunes de programación • Buenas prácticas de programación • Tips de desempeño • Observaciones de ingeniería de software • Tips de prueba y depuración • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
Capítulo 14	Procesamiento de archivos	702
14.1	Introducción	704
14.2	La jerarquía de datos	704
14.3	Archivos y flujos	706
14.4	Creación de un archivo de acceso secuencial	707
14.5	Lectura de datos de un archivo de acceso secuencial	711
14.6	Actualización de archivos de acceso secuencial	717
14.7	Archivos de acceso aleatorio	718
14.8	Creación de un archivo de acceso aleatorio	719
14.9	Escritura aleatoria de datos a un archivo de acceso aleatorio	721
14.10	Lectura secuencial de datos desde un archivo de acceso aleatorio	723
14.11	Ejemplo: un programa de procesamiento de transacciones	725
14.12	Entrada/salida de objetos	730
	<i>Resumen • Terminología • Errores comunes de programación • Buena práctica de programación • Tip de desempeño • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
Capítulo 15	Estructuras de datos	740
15.1	Introducción	742
15.2	Clases autorreferenciadas	743
15.3	Asignación dinámica de memoria	744
15.4	Listas enlazadas	745
15.5	Pilas	758
15.6	Colas	762
15.7	Árboles	766
	<i>Resumen • Terminología • Errores comunes de programación • Buenas prácticas de programación • Tips de desempeño • Tip de portabilidad • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios • Sección especial: construya su propio compilador</i>	
Capítulo 16	Bits, caracteres, cadenas y estructuras	798
16.1	Introducción	800
16.2	Definición de estructuras	800

16.3	Inicialización de estructuras	803
16.4	Uso de estructuras con funciones	803
16.5	typedef	803
16.6	Ejemplo: simulación de barajado y repartición de naipes de alto desempeño	804
16.7	Operadores a nivel de bits	806
16.8	Campos de bits	815
16.9	Biblioteca de manejo de caracteres	818
16.10	Funciones de conversión de cadenas	823
16.11	Funciones de búsqueda de la biblioteca de manejo de cadenas	829
16.12	Funciones de memoria de la biblioteca de manejo de cadenas	833
16.13	Otra función de la biblioteca de manejo de cadenas	837
	<i>Resumen • Terminología • Errores comunes de programación • Buenas prácticas de programación • Tips de desempeño • Tips de portabilidad • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
Capítulo 17	El preprocesador	852
17.1	Introducción	854
17.2	La directiva de preprocesador #include	854
17.3	La directiva de preprocesador #define : constantes simbólicas	855
17.4	La directiva de preprocesador #define : macros	856
17.5	Compilación condicional	858
17.6	Las directivas de preprocesador #error y #pragma	859
17.7	Los operadores # y ##	859
17.8	Números de línea	860
17.9	Constantes simbólicas predefinidas	860
17.10	Aserciones	860
	<i>Resumen • Terminología • Errores comunes de programación • Buena práctica de programación • Tip de desempeño • Observación de ingeniería de software • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
Capítulo 18	Temas sobre el código C heredado	866
18.1	Introducción	868
18.2	Redirección de la entrada/salida en sistemas UNIX y DOS	868
18.3	Listas de argumentos de longitud variable	869
18.4	Uso de argumentos de línea de comandos	871
18.5	Notas sobre la compilación de programas con varios archivos fuente	873
18.6	Terminación del programa con exit y atexit	874
18.7	Calificador de tipo volatile	876
18.8	Sufijos para las constantes enteras y de punto flotante	876
18.9	Manejo de señales	877
18.10	Asignación dinámica de memoria: las funciones calloc y realloc	877
18.11	La bifurcación incondicional: goto	879
18.12	Uniones	881
18.13	Especificaciones de enlace	884

*Resumen • Terminología • Errores comunes de programación • Tips de desempeño
• Tips de portabilidad • Observaciones de ingeniería de software • Ejercicios de auto-
evaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios*

Capítulo 19	La clase <code>string</code> y el procesamiento de flujo de cadenas	892
19.1	Introducción	894
19.2	Asignación y concatenación de objetos tipo <code>string</code>	896
19.3	Comparación de objetos tipo <code>string</code>	898
19.4	Subcadenas	901
19.5	Intercambio de objetos <code>string</code>	901
19.6	Características de los objetos <code>string</code>	902
19.7	Localización de caracteres en un objeto <code>string</code>	904
19.8	Reemplazo de caracteres en un objeto <code>string</code>	906
19.9	Inserción de caracteres en un objeto <code>string</code>	908
19.10	Conversión a cadenas <code>char *</code> estilo C	910
19.11	Iteradores	912
19.12	Procesamiento de flujo de cadenas	913
	<i>Resumen • Terminología • Errores comunes de programación • Buena práctica de programación • Tip de desempeño • Tip de prueba y depuración • Ejercicios de auto- evaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
Capítulo 20	Biblioteca estándar de plantillas (STL)	924
20.1	Introducción	927
20.1.1	Introducción a los contenedores	929
20.1.2	Introducción a los iteradores	933
20.1.3	Introducción a los algoritmos	938
20.2	Contenedores de secuencias	940
20.2.1	El contenedor de secuencia <code>vector</code>	941
20.2.2	El contenedor de secuencia <code>list</code>	949
20.2.3	El contenedor de secuencia <code>deque</code>	953
20.3	Contenedores asociativos	955
20.3.1	El contenedor asociativo <code>multiset</code>	955
20.3.2	El contenedor asociativo <code>set</code>	958
20.3.3	El contenedor asociativo <code>multimap</code>	960
20.3.4	El contenedor asociativo <code>map</code>	962
20.4	Adaptadores de contenedor	964
20.4.1	El adaptador <code>stack</code>	964
20.4.2	El adaptador <code>queue</code>	966
20.4.3	El adaptador <code>priority_queue</code>	968
20.5	Algoritmos	969
20.5.1	<code>fill</code> , <code>fill_n</code> , <code>generate</code> y <code>generate_n</code>	970
20.5.2	<code>equal</code> , <code>mismatch</code> y <code>lexicographical_compare</code>	972
20.5.3	<code>remove</code> , <code>remove_if</code> , <code>remove_copy</code> y <code>remove_copy_if</code>	975
20.5.4	<code>replace</code> , <code>replace_if</code> , <code>replace_copy</code> y <code>replace_copy_if</code>	978

20.5.5	Algoritmos matemáticos	981
20.5.6	Algoritmos básicos de búsqueda y ordenamiento	984
20.5.7	<code>swap</code> , <code>iter_swap</code> y <code>swap_ranges</code>	987
20.5.8	<code>copy_backward</code> , <code>merge</code> , <code>unique</code> y <code>reverse</code>	989
20.5.9	<code>inplace_merge</code> , <code>unique_copy</code> y <code>reverse_copy</code>	991
20.5.10	Operaciones de conjuntos	993
20.5.11	<code>lower_bound</code> , <code>upper_bound</code> y <code>equal_range</code>	996
20.5.12	Heapsort	999
20.5.13	<code>min</code> y <code>max</code>	1002
20.5.14	Algoritmos no tratados en este capítulo	1003
20.6	La clase <code>bitset</code>	1004
20.7	Objetos de función	1008
	<i>Resumen • Terminología • Errores comunes de programación • Buenas prácticas de programación • Tips de desempeño • Tips de portabilidad • Observaciones de ingeniería de software • Tips de prueba y depuración • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios • Recursos de la STL en Internet y en World Wide Web</i>	
Capítulo 21	Adiciones de ANSI/ISO al lenguaje estándar C++	1026
21.1	Introducción	1028
21.2	El tipo de datos <code>bool</code>	1028
21.3	El operador <code>static_cast</code>	1030
21.4	El operador <code>const_cast</code>	1032
21.5	El operador <code>reinterpret_cast</code>	1033
21.6	<code>namespace</code>	1035
21.7	Información de tipo en tiempo de ejecución	1039
21.8	Palabras clave de operador	1043
21.9	Constructores <code>explicit</code>	1044
21.10	Miembros de clase <code>mutable</code>	1049
21.11	Apuntadores a miembros de clase (<code>.*</code> y <code>->*</code>)	1051
21.12	Herencia múltiple y clases base <code>virtual</code>	1053
21.13	Comentarios finales	1057
	<i>Resumen • Terminología • Errores comunes de programación • Buenas prácticas de programación • Tip de desempeño • Tips de portabilidad • Observaciones de ingeniería de software • Tips de prueba y depuración • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
Apéndice A	Tabla de precedencia de los operadores	1064
Apéndice B	Conjunto de caracteres ASCII	1066
Apéndice C	Sistemas numéricos	1067
C.1	Introducción	1068
C.2	Abreviatura de números binarios como números octales y hexadecimales	1071

C.3	Conversión de números octales y hexadecimales a números binarios	1072
C.4	Conversión de binario, octal o hexadecimal a decimal	1072
C.5	Conversión de decimal a binario, octal o hexadecimal	1073
C.6	Números binarios negativos: notación de complemento a dos	1075
	<i>Resumen • Terminología • Ejercicios de autoevaluación • Respuestas a los ejercicios de autoevaluación • Ejercicios</i>	
Apéndice D	Recursos de C++ en Internet y Web	1080
D.1	Recursos	1080
D.2	Tutoriales	1081
D.3	FAQs	1081
D.4	<code>comp.lang.c++</code>	1082
D.5	Compiladores	1083
D.6	Herramientas de desarrollo	1084
Bibliografía		1085
Índice		1092