

Contenido

Prefacio	xxxiii
7 Introducción a las computadoras y a la programación en C++	1
1.1 Introducción	2
1.2 ¿Qué es una computadora?	4
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 máquina, lenguajes ensambladores, y de alto nivel	7
1.7 Historia de C y C++	8
1.8 La Biblioteca estándar de C++	10
1.9 Java	11
1.10 Visual Basic, Visual C++ y C#	11
1.11 Otros lenguajes de alto nivel	13
1.12 Programación estructurada	13
1.13 La tendencia clave del software: tecnología de objetos	14
1.14 Conceptos básicos de un ambiente típico C++	15
1.15 Tendencias de hardware	17
1.16 Historia de Internet	18
1.17 Historia de la World Wide Web	19
1.18 El Consorcio World Wide Web (W3C)	20
1.19 Notas generales acerca de C++ y de este libro	20
1.20 Introducción a la programación en C++	21
1.21 Un programa sencillo: imprimir una línea de texto	21
1.22 Otro programa sencillo: suma de dos enteros	26
1.23 Conceptos de memoria	30
1.24 Aritmética	31
1.25 Toma de decisiones: operadores de igualdad y de relación	34

1.26	Acerca de los objetos: introducción a la tecnología de objetos y al Lenguaje Unificado de Modelado (UML™)	40
1.27	Recorrido a través del libro	44
2	Estructuras de control	70
2.1	Introducción	71
2.2	Algoritmos	72
2.3	Seudocódigo	72
2.4	Estructuras de control	73
2.5	Estructura de selección if	76
2.6	Estructura de selección if/else	77
2.7	Estructura de repetición while	81
2.8	Cómo formular algoritmos: ejemplo práctico 1 (repetición controlada por un contador)	83
2.9	Cómo formular algoritmos mediante el mejoramiento de arriba a abajo, paso a paso: ejemplo práctico 2 (repetición controlada por un centinela)	86
2.10	Cómo formular algoritmos mediante el mejoramiento de arriba a abajo, paso a paso: ejemplo práctico 3 (estructuras de control anidadas)	94
2.11	Operadores de asignación	98
2.12	Operadores de incremento y decremento	99
2.13	Fundamentos de la repetición controlada por un contador	102
2.14	Estructura de repetición for	104
2.15	Ejemplos de la utilización de la estructura for	109
2.16	Estructura de selección múltiple switch	113
2.17	Estructura de repetición do/while	120
2.18	Instrucciones break y continue	122
2.19	Operadores lógicos	124
2.20	La confusión entre los operadores de igualdad (==) y de asignación (=)	127
2.21	Resumen sobre programación estructurada	128
2.22	(Ejemplo práctico opcional) Acerca de los objetos: cómo identificar las clases de un sistema a partir de un problema establecido	133
3	Funciones	169
3.1	Introducción	170
3.2	Componentes de un programa en C++	170
3.3	La biblioteca de funciones matemáticas	171
3.4	Funciones	173
3.5	Definición de funciones	174
3.6	Prototipos de función	178
3.7	Archivos de encabezado	180
3.8	Cómo generar números aleatorios	182
3.9	Ejemplo: un juego de azar y la introducción de enum	188
3.10	Clases para almacenamiento	192
3.11	Reglas de alcance	195
3.12	Recursividad	198
3.13	Ejemplo sobre cómo utilizar la recursividad: la serie de Fibonacci	202
3.14	Recursividad <i>versus</i> iteración	206
3.15	Funciones con listas de parámetros vacías	208

3.16	Funciones inline	209
3.17	Referencias y parámetros por referencia	211
3.18	Argumentos predeterminados	215
3.19	Operador unario de resolución de alcance	217
3.20	Sobrecarga de funciones	219
3.21	Plantillas de función	222
3.22	(Ejemplo práctico opcional) Acerca de los objetos: identificación de los atributos de las clases	225
4	Arreglos	252
4.1	Introducción	253
4.2	Arreglos	253
4.3	Declaración de arreglos	255
4.4	Ejemplos del uso de arreglos	256
4.5	Cómo pasar arreglos a funciones	272
4.6	Ordenamiento de arreglos	276
4.7	Ejemplo práctico: cálculo de la media, la mediana y la moda mediante arreglos	278
4.8	Búsqueda en arreglos: búsqueda lineal y búsqueda binaria	283
4.9	Arreglos con múltiples subíndices	289
4.10	(Ejemplo práctico opcional) Acerca de los objetos: cómo identificar las operaciones de una clase	296
5	Apuntadores y cadenas	319
5.1	Introducción	320
5.2	Declaración e inicialización de variables de apuntador	320
5.3	Operadores para apuntadores	322
5.4	Llamada a funciones por referencia	325
5.5	Uso de const con apuntadores	329
5.6	Ordenamiento burbuja mediante las llamadas por referencia	336
5.7	Expresiones con apuntadores y aritmética de apuntadores	341
5.8	Relación entre apuntadores y arreglos	344
5.9	Arreglos de apuntadores	349
5.10	Ejemplo práctico: simulación para barajar y repartir cartas	350
5.11	Apuntadores a funciones	355
5.12	Introducción al procesamiento de caracteres y texto	360
	5.12.1 Fundamentos de los caracteres y de las cadenas	360
	5.12.2 Funciones para el manejo de cadenas de la biblioteca de manipulación de cadenas	362
5.13	(Ejemplo práctico opcional) Acerca de los objetos: colaboración entre objetos	370
6	Clases y abstracción de datos	404
6.1	Introducción	405
6.2	Definiciones de estructura	406
6.3	Acceso a los miembros de una estructura	407
6.4	Implementación del tipo Tiempo definido por el usuario mediante una estructura al estilo C	408
6.5	Implementación del tipo de dato abstracto Tiempo mediante una clase	411
6.6	Alcance de una clase y acceso a los miembros de la clase	418

6.7	Separación de la interfaz y la implementación	420
6.8	Control de acceso a los miembros	424
6.9	Funciones de acceso y funciones de utilidad	426
6.10	Inicialización de los objetos de una clase: constructores	430
6.11	Uso de argumentos predeterminados con constructores	430
6.12	Destructores	435
6.13	Invocación de constructores y destructores	435
6.14	Uso de las funciones <i>set (establecer)</i> y <i>get (obtener)</i>	439
6.15	Trampa sutil: retorno de una referencia a un dato miembro privado	445
6.16	Asignación predeterminada de miembros	448
6.17	Reutilización de software	450
6.18	(Ejemplo práctico opcional) Acerca de los objetos: Comienzo de la programación de las clases para el simulador del elevador	451
7	Clases: Parte II	468
7.1	Introducción	469
7.2	Objetos y funciones miembro const (constantes)	469
7.3	Composición: objetos como miembros de clases	478
7.4	Funciones y clases friend	485
7.5	Uso del apuntador this	489
7.6	Administración de memoria dinámica con los operadores new y delete	495
7.7	Miembros de clase static	497
7.8	Abstracción de datos y ocultamiento de información	502
7.8.1	Ejemplo: tipo de dato abstracto Arreglo	504
7.8.2	Ejemplo: tipo de dato abstracto Cadena	504
7.8.3	Ejemplo: tipo de dato abstracto Cola	505
7.9	Clases contenedoras e iteradores	505
7.10	Clases proxy	506
7.11	(Ejemplo práctico opcional) Acerca de los objetos: programación de las clases para el simulador del elevador	509
8	Sobrecarga de operadores: objetos de tipo cadena y de tipo arreglo	546
8.1	Introducción	547
8.2	Fundamentos de la sobrecarga de operadores	548
8.3	Restricciones en cuanto a la sobrecarga de operadores	549
8.4	Funciones de operador como miembros de la clase o como funciones friend	550
8.5	Sobrecarga de los operadores de inserción y de extracción de flujo	552
8.6	Sobrecarga de operadores unarios	555
8.7	Sobrecarga de operadores binarios	555
8.8	Ejemplo práctico: la clase Arreglo	556
8.9	Conversión entre tipos	568
8.10	Ejemplo práctico: la clase Cadena	569
8.11	Sobrecarga de ++ y de --	581
8.12	Ejemplo práctico: la clase Fecha	582
8.13	Las clases string y vector de la biblioteca estándar	588

9	Programación orientada a objetos: herencia	609
9.1	Introducción	610
9.2	Clases base y clases derivadas	611
9.3	Miembros <code>protected</code>	614
9.4	Relación entre las clases base y las clases derivadas	614
9.5	Ejemplo práctico: jerarquía de herencia de tres niveles	637
9.6	Constructores, destructores y clases derivadas	642
9.7	Relaciones "Usa un" y "Conoce a"	648
9.8	Herencia <code>public</code> , <code>protected</code> y <code>private</code>	648
9.9	Ingeniería de software con herencia	648
9.10	(Ejemplo práctico opcional) Acerca de los objetos: incorporación de la herencia al simulador del elevador	650
10	Programación orientada a objetos: polimorfismo	662
10.1	Introducción	663
10.2	Relaciones entre los objetos en una jerarquía de herencia	664
10.2.1	Invocación de funciones de clases base a partir de objetos de clases derivadas	665
10.2.2	Dirigir los apuntadores de clases derivadas a objetos de clases base	670
10.2.3	Llamadas a funciones miembro de clases derivadas mediante apuntadores de clases base	672
10.2.4	Funciones virtual	673
10.3	Ejemplos de polimorfismo	679
10.4	Campos de tipos y estructuras <code>switch</code>	680
10.5	Clases abstractas	680
10.6	Caso de estudio: herencia de interfaz y de implementación	682
10.7	Polimorfismo, funciones virtual y vinculación dinámica "tras bambalinas"	695
10.8	Destructores virtuales	699
10.9	Caso de estudio: sistema de nómina utilizando polimorfismo e información de tipos en tiempo de ejecución con <code>dynamic_cast</code> y <code>typeid</code>	699
11	Plantillas	718
11.1	Introducción	719
11.2	Plantillas de funciones	720
11.3	Sobrecarga de plantillas de funciones	723
11.4	Plantillas de clases	723
11.5	Plantillas de clases y parámetros sin tipo	730
11.6	Plantillas y herencia	731
11.7	Plantillas y <code>friends</code>	731
11.8	Plantillas y miembros <code>static</code>	732
12	Entrada/salida de flujo en C++	737
12.1	Introducción	739
12.2	Flujos	739
12.2.1	Comparación entre flujo clásico y flujo estándar	740
12.2.2	Archivos de encabezado de la biblioteca <code>iostream</code>	740
12.2.3	Clases y objetos de entrada/salida de flujo	741

12.3	Salida de flujo	743
12.3.1	Salida de variables tipo <code>char *</code>	743
12.3.2	Salida de caracteres utilizando la función miembro <code>put</code>	744
12.4	Entrada de flujo	744
12.4.1	Las funciones miembro <code>get</code> y <code>getline</code>	745
12.4.2	Las funciones miembro de <code>istream</code> : <code>peek</code> , <code>putback</code> e <code>ignore</code>	748
12.4.3	E/S con seguridad de tipos	748
12.5	E/S sin formato utilizando <code>read</code> , <code>write</code> y <code>gcount</code>	748
12.6	Manipuladores de flujo	749
12.6.1	Base de flujo integral: <code>dec</code> , <code>oct</code> , <code>hex</code> y <code>setbase</code>	750
12.6.2	Precisión de punto flotante (<code>precision</code> , <code>setprecision</code>)	751
12.6.3	Anchura de campos (<code>width</code> , <code>setw</code>)	752
12.6.4	Manipuladores definidos por el programador	754
12.7	Estados de formato de flujo y manipuladores de flujo	755
12.7.1	Ceros a la derecha y puntos decimales (<code>showpoint</code>)	756
12.7.2	Alineación (<code>left</code> , <code>right</code> e <code>internal</code>)	757
12.7.3	Relleno (<code>fill</code> , <code>setfill</code>)	759
12.7.4	Base integral de flujo (<code>dec</code> , <code>oct</code> , <code>hex</code> , <code>showbase</code>)	760
12.7.5	Números de punto flotante; notación científica y fija (<code>scientific</code> , <code>fixed</code>)	761
12.7.6	Control de mayúsculas/minúsculas (<code>uppercase</code>)	762
12.7.7	Especificación de formato booleano (<code>boolalpha</code>)	763
12.7.8	Establecer y restablecer el estado del formato mediante la función miembro <code>flags</code>	764
12.8	Estados de error de flujo	766
12.9	Enlazar un flujo de salida a un flujo de entrada	768
13	Manejo de excepciones	779
13.1	Introducción	780
13.2	Generalidades acerca del manejo de excepciones	781
13.3	Otras técnicas para el manejo de errores	783
13.4	Ejemplo simple del manejo de excepciones: dividir entre cero	784
13.5	Volver a lanzar una excepción	788
13.6	Especificaciones de excepciones	789
13.7	Procesamiento de excepciones inesperadas	790
13.8	Limpieza de la pila	790
13.9	Constructores, destructores y manejo de excepciones	792
13.10	Excepciones y herencia	793
13.11	Procesamiento de fallas con <code>new</code>	793
13.12	La clase <code>auto_ptr</code> y la asignación dinámica de memoria	797
13.13	Jerarquía de excepciones de la biblioteca estándar	800
14	Procesamiento de archivos	808
14.1	Introducción	809
14.2	Jerarquía de datos	809
14.3	Archivos y flujos	811
14.4	Creación de un archivo de acceso secuencial	812
14.5	Cómo leer datos de un archivo de acceso secuencial	816

14.6	Actualización de archivos de acceso secuencial	823
14.7	Archivos de acceso aleatorio	824
14.8	Creación de un archivo de acceso aleatorio	824
14.9	Cómo escribir datos en forma aleatoria en un archivo de acceso aleatorio	829
14.10	Cómo leer datos secuencialmente desde un archivo de acceso aleatorio	831
14.11	Ejemplo: un programa para procesar transacciones	834
14.12	Entrada/salida de objetos	841

15 La clase string y el procesamiento de flujos de cadena **850**

15.1	Introducción	851
15.2	Asignación y concatenación de objetos tipo <code>string</code>	852
15.3	Comparación de objetos tipo <code>string</code>	855
15.4	Subcadenas	857
15.5	Intercambio de objetos <code>string</code>	858
15.6	Características de la clase <code>string</code>	859
15.7	Buscar cadenas y caracteres en un objeto <code>string</code>	862
15.8	Sustitución de caracteres en un objeto <code>string</code>	864
15.9	Inserción de caracteres en un objeto <code>string</code>	866
15.10	Conversión a cadenas <code>char *</code> estilo C	867
15.11	Iteradores	869
15.12	Procesamiento de flujos de cadena	870

16 Programación Web con CGI **880**

16.1	Introducción	881
16.2	Tipos de peticiones HTTP	882
16.3	Arquitectura de multinivel	882
16.4	Acceso a los servidores Web	883
16.5	El servidor HTTP Apache	884
16.6	Cómo pedir documentos de XHTML	885
16.7	Introducción a CGI	885
16.8	Transacción HTTP simple	886
16.9	Secuencia de comandos CGI simple	888
16.10	Envío de datos de entrada a una secuencia de comandos CGI	895
16.11	Uso de formularios en XHTML para enviar datos de entrada	897
16.12	Otros encabezados	905
16.13	Ejemplo práctico: una página Web interactiva	905
16.14	Cookies	909
16.15	Archivos del lado del servidor	915
16.16	Ejemplo práctico: carrito de compras	921
16.17	Recursos en Internet y Web	936

17 Estructuras de datos **942**

17.1	Introducción	943
17.2	Clases autorreferenciadas	944
17.3	Asignación de memoria y estructuras de datos dinámicas	945
17.4	Listas enlazadas	945
17.5	Pilas	960

17.6	Colas	965
17.7	Árboles	969
18	Bits, caracteres, cadenas y estructuras	1000
18.1	Introducción	1001
18.2	Definición de estructuras	1001
18.3	Inicialización de estructuras	1003
18.4	Uso de estructuras con funciones	1004
18.5	typedef	1004
18.6	Ejemplo: simulación de alto rendimiento para barajar y repartir cartas	1005
18.7	Operadores a nivel de bits	1007
18.8	Campos de bits	1017
18.9	Biblioteca para manejo de caracteres	1020
18.10	Funciones de conversión de cadenas	1026
18.11	Funciones de búsqueda de la biblioteca para manejo de cadenas	1031
18.12	Funciones de memoria de la biblioteca para manejo de cadenas	1036
19	El preprocesador	1053
19.1	Introducción	1054
19.2	La directiva #include del preprocesador	1054
19.3	La directiva #define del preprocesador: constantes simbólicas	1055
19.4	La directiva #define del preprocesador: macros	1056
19.5	Compilación condicional	1057
19.6	Las directivas #error y #pragma del preprocesador	1058
19.7	Los operadores # y ##	1059
19.8	Números de línea	1059
19.9	Constantes simbólicas predefinidas	1060
19.10	Afirmaciones	1060
20	Temas relacionados con el código heredado de C	1065
20.1	Introducción	1066
20.2	Redirección de la entrada/salida en sistemas UNIX y DOS	1066
20.3	Listas de argumentos con longitudes variables	1067
20.4	Uso de los argumentos de la línea de comandos	1070
20.5	Notas sobre la compilación de programas con varios archivos de código fuente	1071
20.6	Terminación de un programa con exit y atexit	1073
20.7	Calificador de tipo volatile	1075
20.8	Sufijos para constantes enteras y de punto flotante	1075
20.9	Manejo de señales	1075
20.10	Asignación dinámica de memoria con calloc y realloc	1078
20.11	La ramificación incondicional: goto	1079
20.12	Uniones	1080
20.13	Especificaciones de enlazado	1084
21	Biblioteca estándar de plantillas (STL)	1090
21.1	Introducción a la Biblioteca estándar de plantillas (STL)	1092
21.1.1	Introducción a los contenedores	1094
21.1.2	Introducción a los iteradores	1098

21.1.3	Introducción a los algoritmos	1103
21.2	Contenedores de secuencia	1105
21.2.1	Contenedor de secuencia <code>vector</code>	1105
21.2.2	Contenedor de secuencia <code>list</code>	1113
21.2.3	Contenedor de secuencia <code>deque</code>	1117
21.3	Contenedores asociativos	1119
21.3.1	Contenedor asociativo <code>multiset</code>	1119
21.3.2	Contenedor asociativo <code>set</code>	1122
21.3.3	Contenedor asociativo <code>multimap</code>	1124
21.3.4	Contenedor asociativo <code>map</code>	1126
21.4	Adaptadores de contenedores	1128
21.4.1	Adaptador <code>stack</code>	1128
21.4.2	Adaptador <code>queue</code>	1130
21.4.3	Adaptador <code>priority_queue</code>	1132
21.5	Algoritmos	1133
21.5.1	<code>fill</code> , <code>fill_n</code> , <code>generate</code> y <code>generate_n</code>	1134
21.5.2	<code>equal</code> , <code>mismatch</code> y <code>lexicographical_compare</code>	1136
21.5.3	<code>remove</code> , <code>remove_if</code> , <code>remove_copy</code> y <code>remove_copy_if</code>	1138
21.5.4	<code>replace</code> , <code>replace_if</code> , <code>replace_copy</code> y <code>replace_copy_if</code>	1141
21.5.5	Algoritmos matemáticos	1144
21.5.6	Algoritmos básicos de búsqueda y ordenamiento	1148
21.5.7	<code>swap</code> , <code>iter_swap</code> y <code>swap_ranges</code>	1150
21.5.8	<code>copy_backward</code> , <code>merge</code> , <code>unique</code> y <code>reverse</code>	1152
21.5.9	<code>inplace_merge</code> , <code>unique_copy</code> y <code>reverse_copy</code>	1154
21.5.10	Operaciones de conjuntos	1156
21.5.11	<code>lower_bound</code> , <code>upper_bound</code> y <code>equal_range</code>	1160
21.5.12	Heapsort	1162
21.5.13	<code>min</code> y <code>max</code>	1165
21.5.14	Algoritmos que no se cubren en este capítulo	1166
21.6	La clase <code>bitset</code>	1168
21.7	Objetos de función	1172
21.8	Recursos de la STL en Internet y la Web	1175
22	Otros temas	1183
22.1	Introducción	1184
22.2	El operador <code>const_cast</code>	1184
22.3	El operador <code>reinterpret_cast</code>	1185
22.4	Espacios de nombres	1186
22.5	Palabras reservadas de los operadores	1190
22.6	Constructores <code>explicit</code>	1192
22.7	Miembros de clases <code>mutable</code>	1197
22.8	Apuntadores a miembros de clases (<code>.*</code> y <code>->*</code>)	1199
22.9	Herencia múltiple	1201
22.10	Herencia múltiple y clases base <code>virtual</code>	1205
22.11	Comentarios finales	1210
A	Tabla de precedencia de los operadores	1214

B	Conjunto de caracteres ASCII	1216
C	Sistemas numéricos	1217
C.1	Introducción	1218
C.2	Abreviatura de los números binarios como números octales y hexadecimales	1221
C.3	Conversión de números octales y hexadecimales a números binarios	1222
C.4	Conversión de un número binario, octal o hexadecimal a decimal	1222
C.5	Conversión de un número decimal a binario, octal o hexadecimal	1223
C.6	Números binarios negativos: notación de complemento a dos	1225
D	Recursos de C++ en Internet y Web	1230
D.1	Recursos	1230
D.2	Tutoriales	1232
D.3	FAQs	1233
D.4	Visual C++	1233
D.5	Grupos de noticias	1233
D.6	Compiladores y herramientas de desarrollo	1234
D.7	Biblioteca de plantillas estándar	1234
E	Introducción a XHTML	1236
E.1	Introducción	1237
E.2	Edición de XHTML	1237
E.3	El primer ejemplo de XHTML	1238
E.4	Encabezados	1240
E.5	Vínculos	1242
E.6	Imágenes	1245
E.7	Caracteres especiales y más interrupciones de línea	1249
E.8	Listas desordenadas	1250
E.9	Listas anidadas y ordenadas	1251
E.10	Tablas básicas de XHTML	1252
E.11	Tablas intermedias de XHTML y formato	1257
E.12	Formularios básicos de XHTML	1259
E.13	Formularios más complejos de XHTML	1262
E.14	Recursos en Internet y en World Wide Web	1269
F	Caracteres especiales XHTML	1274
	Bibliografía	1275
	Índice	1281