

INDICE

1. Introducción	1
1.1. Desarrollo de sistemas y programación	2
1.2. ¿Qué es el software?	2
1.3. Documentación en programación	4
1.4. Lo que posiblemente sea el programa completo más sencillo posible en C++	6
1.5. Disposición del código y comentarios	7
1.6. Producción y ejecución de un ejecutable	11
2. Abstracción en la Programación	13
2.1. Modularidad y abstracción	13
2.2. Especificación e implementación	15
2.3. Tipos de datos y apoyo para la abstracción	15
2.4. Un ejemplo de abstracción funcional	16
2.5. El proceso de diseño e implementación	18
2.6. Bibliotecas y entrada/salida	21
3. Los Tipos de Datos Primitivos	23
3.1. Literales y variables	23
3.1.1. Nombres de las variables	24
3.1.2. Asignación de valores iniciales a las variables	26
3.2. Los tipos primitivos: valores y definiciones	26
3.2.1. Números enteros	27
3.2.2. Caracteres	32
3.2.3. Números reales	34
3.3. Entrada/Salida de tipos de datos primitivos	37
4. Primeros Pasos de la Construcción: Expresiones	39
4.1. Evaluación de expresiones	40
4.2. Operaciones y expresiones	41
4.3. Operadores aritméticos	41
4.3.1. Operadores unarios	41
4.3.2. Operadores binarios	42
4.3.3. Un ejemplo de programa	43
4.4. Operadores de manipulación de bits	44
4.4.1. Operadores unarios	45
4.4.2. Operadores binarios	45
4.4.3. Un ejemplo de programa	46
4.5. Operadores relacionales	47
4.6. Operadores de asignación	49
4.6.1. Asignación simple	50
4.6.2. Asignación compuesta	51
4.6.3. Operadores de incremento y decremento	53
4.7. Evaluación de expresiones: prioridad y asociatividad	55
4.8. Expresiones y conversiones de tipos	56
4.8.1. Conversión de tipos explícita	58
4.8.2. Conversión de tipos implícita	58
4.9. Variables constantes (constantes con nombre)	60
5. Control de Flujo: Secuencia, Toma de Decisiones e Iteración	65
5.1. Secuencia	66

5.1.1. Declaraciones y definiciones	66
5.1.2. Sentencias que son expresiones	68
5.1.3. La sentencia nula	69
5.1.4. Sentencias compuestas	69
5.2. Toma de decisiones	70
5.2.1. Selección binaria	70
5.2.2. La expresión condicional	75
5.2.3. Selección múltiple	76
5.3. Iteración	83
5.3.1. Iteración con comprobación al principio	84
5.3.2. Iteración con comprobación al final	93
5.3.3. Iteración con comprobación intermedia	95
5.3.4. Bucles infinitos genuinos	97
5.4. Ausencia de control de flujo	98
6. Abstracción Funcional	101
6.1. Declaración y definición de funciones	102
6.2. Representación de funciones mediante diagramas	103
6.3. Tipos de retorno y valores de retorno	104
6.3.1. Asignación de valores de retorno: La sentencia return	105
6.3.2. Funciones void	108
6.4. Paso de parámetros	108
6.4.1. Paso por valor	109
6.4.2. Paso por referencia	110
6.4.3. Referencias y valores de retorno	116
6.5. Valores de los parámetros por omisión	117
6.6. Número variable de parámetros	118
6.7. Funciones inline	120
6.8. Desarrollo de funciones	121
6.9. Ámbito, visibilidad y duración	132
6.9.1. Cosas globales y cosas locales	133
6.9.2. Ámbito	133
6.9.3. Visibilidad	135
6.9.4. Duraciones	137
6.9.5. Consecuencias para la programación	139
6.10. Un añadido al control de flujo: la recursividad	140
6.11. Ordenes de crecimiento de los recursos	147
7. Una Estructura de datos Primitiva: Las Matrices (arrays)	151
7.1. Matrices unidimensionales	152
7.2. Asignación de valores iniciales a matrices	156
7.3. Límites de las matrices	158
7.4. Las matrices como parámetros de una función	159
7.5. Un ejemplo que utiliza matrices: ordenación del contenido de una matriz	161
7.6. Matrices de caracteres: cadenas (strings)	169
7.6.1. Cadenas literales	169
7.6.2. Asignación de valores iniciales a cadenas	171
7.6.3. Entrada/salida de cadenas	172
7.7. Matrices multidimensionales	176

8. Tipos de Daos Definidos por el Usuario: Clases	179
8.1. Definición de clases	180
8.2. Declaración de clases	181
8.3. Apoyo para la abstracción: información pública y privada	182
8.4. Una brecha en el sistema de protección: los amigos (friends)	184
8.5. Acceso a los miembros de una clase	185
8.6. Implementación de funciones miembro (métodos)	187
8.7. Constructores y destructores	189
8.7.1. Constructores	190
8.7.2. Destructores	199
8.8. Valores y literales	199
8.9. Asignación de valores iniciales a matrices de clases	201
8.10. Funciones miembro inline	202
8.11. Constantes con nombre y funciones miembro constantes	204
8.12. Alusiones a la propia variable	205
8.13. Enumeraciones, segunda parte	206
9. Una Forma de Polimorfismo: La Sobrecarga	213
9.1. Sobrecarga de funciones	215
9.2. Uso de funciones miembro para obtener polimorfismo	216
9.3. Sobrecarga de operadores	216
9.3.1. Restricciones de la sobrecarga de operadores	219
9.3.2. Sobrecarga y cambio de tipos	221
9.3.3. Estilos de sobrecarga de operadores	225
9.3.4. Sobrecarga del operador de indexación	227
9.3.5. Sobrecarga del operador llamada a función ()	230
9.3.6. Sobrecarga de la asignación	230
9.3.7. Sobrecarga de los operadores de incremento y decremento	232
9.4. La biblioteca IOSTREAM y la sobrecarga	234
9.5. Una clase de números complejos	243
9.6. Tipos de subintervalo	249
10. Un Poco de Dinamismo: Los Tipos Puntero	255
10.1. ¿Qué es un puntero?	256
10.2. Declaración y definición de punteros	257
10.3. Sinónimos de tipos	260
10.4. Punteros a constantes	263
10.5. Punteros constantes	263
10.6. Opciones con punteros, aritmética de punteros y matrices	264
10.7. Punteros y matrices como parámetros	270
10.8. Los punteros como valores devueltos	272
10.9. Cadenas (de nuevo)	275
10.9.1. Asignación de valores iniciales a variables alfanuméricas	275
10.9.2. Entrada/Salida de cadenas, II	277
10.9.3. Parámetros de la línea de órdenes	278
10.10. Punteros a tipos definidos por el usuario	284
10.11. Punteros a funciones	286
10.12. Punteros a miembros de una clase	286
10.13. Literales de punteros	303
11. Tipos de Datos Dinámicos	305

11.1. El operador new	306
11.2. El operador delete	308
11.3. Cuando se agota la memoria	311
11.4. Matrices dinámicas	316
11.5. Listas enlazadas	318
11.6. Utilización de listas enlazadas para crear abstracciones	326
11.6.1. Lista ordenadas	326
11.6.2. Pilas	346
11.6.3. Colas	354
11.7. ejemplo de árboles binarios: un léxico	359
11.7.1. Aproximación TAD	363
11.7.2. Una aproximación más orientada a objetos	376
11.8. Reserva dinámica de espacio para una clase específica	381
11.8.1. Miembros estáticos de una clase	382
11.8.2. Mantenimiento de listas libres	384
12. Polimorfismo: Tipos Genéricos	391
12.1. Plantillas de funciones	392
12.2. Plantillas de clases	402
12.2.1. Punteros como elementos de datos	408
12.2.2. Tipos de punteros definidos por el usuario	410
12.2.3. Argumentos formales sin tipo en las plantillas	421
12.3. Colecciones heterogéneas	423
12.3.1. Uniones	423
12.3.2. Uniones anónimas	426
12.3.3. Uso de las uniones	427
13. Abstracciones que se pueden Extender: Herencia	431
13.1. La jerga de la herencia	433
13.2. Definición de subclases	433
13.3. Polimorfismo inminente	439
13.4. Control de la herencia	440
13.4.1. Herencia y ámbito	440
13.4.2. Tipos de herencia	441
13.4.3. Herencia y friends	444
13.5. Herencia múltiple	444
13.6. Utilización de :: para acceder a variables y funciones	449
13.7. Punteros y referencias a tipos que implica herencia	451
13.8. Funciones virtuales	458
13.9. Funciones virtuales abstractas y clases abstractas	462
13.10. La herencia como restricción de acceso	469
13.11. Herencia y reutilización del código	472
13.12. Herencia y tipos discriminados	478
13.13. Programación orientado a objetos	490
14. Archivos: Cosas de Utilidad General	499
14.1. Estructura de un programa, II: Archivos y control de la compilación	499
14.1.1. El proceso de la compilación	500
14.1.2. Control de la compilación	502
14.1.3. Ámbito y archivos	512
14.1.4. Bibliotecas	513

14.1.5. Automatización del control de compilación	516
14.2. Los archivos como depósito de datos	517
14.2.1. Conceptos básicos de la gestión de archivos	518
14.2.2. Entrada/salida en IOSTREAM	519
14.2.3. Procesamiento de archivos	521
14.2.4. Filtros	532
15. De Todo un Poco	535
15.1. Manipulación de bits	535
15.1.1. Ahorrando espacio	536
15.1.2. Gestión del hardware	537
15.1.3. Campos de bits	538
15.1.4. Volatile	541
15.2. Optimización	542
15.2.1. Seguimiento del código	543
15.2.2. Variables de registro	544
15.2.3. Uso de código en ensamblador	546
15.3. Punto final	546
Apéndice A. Bibliografía	549
A.1. C++ como lenguaje	549
A.2. Programación orientada a objetos	550
A.3. Programación en Scheme (Lisp)	551
A.4. Algoritmos	551
A.5. Ingeniería del software	551
A.6. Interacción humano – computadora	552
A.7. Especificación de sistemas	553
A.8. Matemáticas	553
A.9. Hardware de computadoras	554
Apéndice B. Soluciones a los Ejercicios	555
Ejercicio 3.1.- Pagina 37	555
Ejercicio 3.2.- Página 38	557
Ejercicio 4.1.- Página 44	558
Ejercicio 4.3.- Pagina 54	558
Ejercicio 5.1.- Pagina 73	559
Ejercicio 5.2.- Pagina 100	561
Ejercicio 5.3.- Pagina 100	563
Ejercicio 6.1.- Pagina 112	565
Ejercicio 6.2.- Pagina 112	566
Ejercicio 6.3.- Pagina 132	566
Ejercicio 6.4.- Pagina 142	568
Ejercicio 6.5.- Pagina 149	569
Ejercicio 6.6.- Pagina 150	571
Ejercicio 7.1.- Pagina 167	571
Ejercicio 7.2.- Pagina 168	571
Ejercicio 8.1.- Pagina 196	573
Ejercicio 8.2.- Pagina 206	574
Ejercicio 9.1.- Pagina 230	576
Ejercicio 9.2.- Pagina 242	578
Ejercicio 9.3.- Pagina 249	581

Ejercicio 9.4.- Pagina 254	590
Ejercicio 9.5.- Pagina 254	592
Ejercicio 10.1.- Pagina 260	599
Ejercicio 10.2.- Pagina 264	599
Ejercicio 10.3.- Pagina 264	600
Ejercicio 10.4.- Pagina 269	600
Ejercicio 10.5.- Pagina 281	601
Ejercicio 10.6.- Pagina 304	602
Ejercicio 11.1.- Pagina 318	607
Ejercicio 11.2.- Pagina 341	612
Ejercicio 11.3.- Pagina 346	614
Ejercicio 11.4.- Pagina 346	620
Ejercicio 11.5.- Pagina 352	626
Ejercicio 11.6.- Pagina 353	628
Ejercicio 11.7.- Pagina 354	628
Ejercicio 11.8.- Pagina 359	630
Ejercicio 11.9.- Pagina 368	635
Ejercicio 11.10.- Pagina 375	641
Ejercicio 11.11.- Pagina 381	645
Ejercicio 11.12.- Pagina 381	650
Ejercicio 11.13.- Pagina 381	650
Ejercicio 11.14.- Pagina 383	650
Ejercicio 12.1.- Pagina 407	651
Ejercicio 12.2.- Pagina 418	657
Ejercicio 12.3. - Pagina 421	658
Ejercicio 13.1. Pagina 478	668
Ejercicio 13.2.- Pagina 497	669
Ejercicio 14.1.- Pagina 531	675
Índice	683