
Sommario

PREFAZIONE	XI
Lo scopo di questo libro	xi
Le sezioni “Pensare in termini di oggetti”	xii
Il CD-ROM	xiv
La metodologia di insegnamento	xiv
L'apprendimento attraverso il codice	xiv
L'accesso al World Wide Web	xiv
Obiettivi	xiv
Il codice e gli esempi	xv
La programmazione orientata agli oggetti	xv
Figure e immagini	xv
Consigli e suggerimenti	xv
Esercizi di autovalutazione	xv
Esercizi	xvi
Indice analitico	xvi
Panoramica del libro	xvi
CAPITOLO I: INTRODUZIONE: I COMPUTER, LA PROGRAMMAZIONE E IL C++	I
1.1 Introduzione	1
1.2 Che cos'è un computer?	3
1.3 La struttura del computer	4
1.4 L'evoluzione dei sistemi operativi	5
1.5 I personal computer, i sistemi distribuiti e i sistemi client/server	6
1.6 I linguaggi macchina, assembly e ad alto livello	6
1.7 Il C e il C++: un po' di storia	8
1.8 La libreria standard del C++	9
1.9 Java, Internet e il World Wide Web	10
1.10 Altri linguaggi ad alto livello	10
1.11 La programmazione strutturata	11
1.12 Gli elementi fondamentali di un tipico ambiente C++	11
1.13 Alcune considerazioni generali sul C++ e sul nostro corso	14
1.14 Un programma semplice: visualizzare una linea di testo	16
1.15 Un altro semplice programma: l'addizione di due numeri interi	20
1.16 La memoria: concetti fondamentali	24
1.17 I calcoli aritmetici	25
1.18 Prendere decisioni: gli operatori relazionali e di uguaglianza	28
1.19 Le nuove convenzioni per i file di intestazione e gli spazi dei nomi	32
1.20 Pensare in termini di oggetti: le tecniche orientate agli oggetti e UML (Unified Modeling Language™)	34
Introduzione all'analisi e alla progettazione orientate agli oggetti (OOAD)	36
Storia di UML	37
Che cos'è UML?	38

Esercizi di autovalutazione.....	38
Risposte agli esercizi di autovalutazione	40
Esercizi	42
CAPITOLO 2: LE STRUTTURE DI CONTROLLO	47
2.1 Introduzione	47
2.2 Gli algoritmi	47
2.3 Lo pseudocodice	48
2.4 Le strutture di controllo	48
2.5 La struttura di selezione if	51
2.6 La struttura di selezione if/else	53
2.7 La struttura iterativa while	57
2.8 Tipologie degli algoritmi di iterazione: l'iterazione controllata da un contatore	58
2.9 Tipologie degli algoritmi di iterazione: l'iterazione controllata da un valore sentinella	60
2.10 Tipologie di algoritmi e ridefinizione top-down: le strutture di controllo nidificate	67
2.11 Gli operatori di assegnamento	71
2.12 Gli operatori di incremento e decremento	72
2.13 Concetti fondamentali dei cicli controllati da variabili contatore	75
2.14 La struttura di iterazione for	77
2.15 Esempi di costrutti for	81
2.16 La struttura di selezione switch	85
2.17 La struttura iterativa do/while	91
2.18 Le istruzioni break e continue	93
2.19 Gli operatori logici	95
2.20 Un errore tipico: confondere l'operatore di uguaglianza == con l'operatore di assegnamento =	98
2.21 Riepilogo dei concetti fondamentali della programmazione strutturata	100
2.22 Pensare in termini di oggetti: come individuare le classi in un problema [progetto opzionale]	106
Definizione del problema	106
Analisi e progettazione del sistema	110
I diagrammi dei casi d'uso	111
L'identificazione delle classi di un sistema	112
I diagrammi delle classi	113
I diagrammi degli oggetti	116
Esercizi di autovalutazione.....	117
Risposte agli esercizi di autovalutazione	119
Esercizi	122
CAPITOLO 3: LE FUNZIONI	133
3.1 Introduzione	133
3.2 I componenti di un programma in C++	133
3.3 Le funzioni matematiche della libreria standard	135
3.4 Le funzioni	136

3.5	La definizione di una funzione	137
3.6	I prototipi di funzione	141
3.7	I file di intestazione	144
3.8	La generazione di numeri casuali	146
3.9	I giochi d'azzardo e la parola riservata enum	151
3.10	Le informazioni di memorizzazione	155
3.11	Le regole di visibilità	157
3.12	Il concetto di ricorsione	161
3.13	Un altro esempio di ricorsione: la serie di Fibonacci	164
3.14	Ricorsione o iterazione?	168
3.15	Le funzioni che hanno una lista di parametri vuota	170
3.16	Le funzioni in linea	171
3.17	I riferimenti e il passaggio di parametri per riferimento	173
3.18	Gli argomenti di default	177
3.19	L'operatore unario di risoluzione dello scope	179
3.20	L'overloading delle funzioni	180
3.21	Le funzioni generiche	182
3.22	Pensare in termini di oggetti: come identificare gli attributi di una classe [progetto opzionale]	184
	I diagrammi di stato	186
	I diagrammi delle attività	188
	Conclusione	190
	Esercizi di autovalutazione	190
	Risposte agli esercizi di autovalutazione	193
	Esercizi	195
CAPITOLO 4: GLI ARRAY		205
4.1	Introduzione	205
4.2	Gli array	205
4.3	Come si dichiara un array	208
4.4	Alcuni esempi di array	208
4.5	Il passaggio di un array a una funzione	222
4.6	Gli algoritmi di ordinamento e gli array	227
4.7	Il calcolo di media, mediana e moda con gli array	229
4.8	Le ricerche in array: ricerca lineare e binaria	233
4.9	Gli array multidimensionali	238
4.10	Pensare in termini di oggetti: come identificare le operazioni di una classe [progetto opzionale]	244
	I diagrammi di sequenza	249
	Esercizi di autovalutazione	252
	Risposte agli esercizi di autovalutazione	253
	Esercizi	254
	Esercizi sulla ricorsione	264
CAPITOLO 5: PUNTATORI E STRINGHE		265
5.1	Introduzione	265
5.2	Come si dichiarano e si inizializzano i puntatori	265

5.3	Gli operatori di manipolazione dei puntatori	267
5.4	La chiamata per riferimento con argomenti di tipo puntatore	269
5.5	Privilegi di accesso e passaggio dei parametri	274
5.6	L'algoritmo di ordinamento a bolle con i puntatori	279
5.7	L'aritmetica dei puntatori	285
5.8	La correlazione tra puntatori e array	287
5.9	Gli array di puntatori	292
5.10	Un programma per mescolare e distribuire carte da gioco	293
5.11	I puntatori a funzione	298
5.12	Introduzione alla manipolazione di caratteri e stringhe	302
	Caratteri e stringhe: concetti fondamentali	303
	Le funzioni di libreria per le stringhe	305
5.13	Pensare in termini di oggetti: le collaborazioni tra gli oggetti	
	[progetto opzionale]	312
	I diagrammi delle collaborazioni	314
	Riepilogo	315
	Risorse in rete sull'UML.....	316
	Esercizi di autovalutazione.....	320
	Risposte agli esercizi di autovalutazione	322
	Esercizi	323
	Sezione speciale: costruite il vostro computer	327
	Ulteriori esercizi sui puntatori	332
	Esercizi sulla manipolazione di stringhe	336
	Sezione speciale: esercizi avanzati sulla manipolazione di stringhe	338
	Un progetto di manipolazione di stringhe complesso	341
CAPITOLO 6: LE CLASSI E L'ASTRAZIONE DEI DATI		343
6.1	Introduzione	343
6.2	Come si definisce una struttura	344
6.3	Come si accede ai membri di una struttura	345
6.4	L'implementazione del tipo di dato Time come struttura	346
6.5	L'implementazione del tipo di dato Time come classe	348
6.6	La visibilità a livello di classe e l'accesso ai membri di una classe	356
6.7	La separazione di interfaccia e implementazione	357
6.8	Il controllo dell'accesso ai membri di una classe	361
6.9	Le funzioni di accesso e di utilità	364
6.10	L'inizializzazione degli oggetti di una classe: i costruttori	367
6.11	I costruttori e gli argomenti di default	368
6.12	I distruttori	371
6.13	Quando sono chiamati i costruttori e i distruttori?	372
6.14	L'utilizzo dei dati e delle funzioni membro	375
6.15	Un sottile errore logico: restituire un riferimento	
	a un dato membro private	380
6.16	L'assegnamento tra oggetti: la copia di default membro a membro	382
6.17	Ancora sul concetto di software riutilizzabile	384
6.18	Pensare in termini di oggetti: programmazione delle classi del simulatore	
	[progetto opzionale]	384

Implementazione: visibilità	385
Implementazione: handle	386
Implementazione: i file di intestazione delle classi	386
Conclusione	395
Esercizi di autovalutazione	395
Risposte agli esercizi di autovalutazione	396
Esercizi	396
CAPITOLO 7: LE CLASSI: SECONDA PARTE	399
7.1 Introduzione	399
7.2 Gli oggetti e le funzioni membro costanti	399
7.3 Il concetto di composizione: oggetti che diventano membri di altre classi	408
7.4 Le funzioni e le classi friend	413
7.5 Il puntatore this	416
7.6 L'allocazione dinamica della memoria: gli operatori new e delete	421
7.7 I membri static di una classe	423
7.8 Due concetti importanti: astrazione dei dati e occultamento delle informazioni	429
Il tipo di dato astratto "array"	430
Il tipo di dato astratto "stringa"	431
Il tipo di dato astratto "coda"	431
7.9 Le classi container e gli iteratori	432
7.10 Le classi proxy	432
7.11 Pensare in termini di oggetti: l'implementazione delle classi del simulatore [progetto opzionale]	435
Una panoramica dell'implementazione	435
Implementazione del simulatore	436
Esercizi di autovalutazione	462
Risposte agli esercizi di autovalutazione	463
Esercizi	464
CAPITOLO 8: L'OVERLOADING DEGLI OPERATORI	467
8.1 Introduzione	467
8.2 L'overloading degli operatori: concetti fondamentali	468
8.3 Restrizioni	470
8.4 La progettazione delle funzioni operatore: funzioni membro o funzioni friend?	471
8.5 L'overloading degli operatori di inserimento/estrazione per l'I/O su stream	473
8.6 L'overloading degli operatori unari	476
8.7 L'overloading degli operatori binari	476
8.8 Progettazione della classe Array	477
8.9 Conversioni tra tipi diversi	489
8.10 Progettazione della classe String	490
8.11 L'overloading degli operatori ++ e --	502

8.12	Progettazione della classe Date	503
	Esercizi di autovalutazione.....	508
	Risposte agli esercizi di autovalutazione	508
	Esercizi	509
CAPITOLO 9: L'EREDITARIETÀ		517
9.1	Introduzione	517
9.2	Le classi base e le classi derivate.....	519
9.3	I membri protected	521
9.4	Il cast dei puntatori a una classe base in puntatori a una classe derivata	521
9.5	Utilizzo delle funzioni membro	527
9.6	L'overriding di membri della classe base in una classe derivata	528
9.7	Ereditarietà di tipo public, protected e private	532
9.8	Classi base dirette e indirette	533
9.9	Utilizzo dei costruttori e dei distruttori nelle classi derivate	534
9.10	Conversione implicita di un oggetto di una classe derivata in oggetto della classe base	537
9.11	Il ruolo dell'ereditarietà nell'ingegneria del software	539
9.12	Composizione ed ereditarietà	540
9.13	Le relazioni "uses a" e "knows a"	540
9.14	Progettazione delle classi Point, Circle e Cylinder	541
9.15	L'ereditarietà multipla	548
9.16	Pensare in termini di oggetti: come sfruttare l'ereditarietà nel simulatore di ascensore [progetto opzionale]	553
	Esercizi di autovalutazione.....	559
	Risposte agli esercizi di autovalutazione	560
	Esercizi	560
CAPITOLO 10: LE FUNZIONI VIRTUALI E IL POLIMORFISMO		563
10.1	Introduzione	563
10.2	I campi di tipo e le istruzioni switch	563
10.3	Le funzioni virtuali	564
10.4	Le classi base astratte e le classi concrete	565
10.5	Il polimorfismo	566
10.6	Progettazione di un libro paga elettronico	568
10.7	L'aggiunta di nuove classi e il binding dinamico	579
10.8	I distruttori virtuali.....	579
10.9	L'ereditarietà di interfaccia e di implementazione.....	580
10.10	L'implementazione di polimorfismo, funzioni virtuali e binding dinamico	588
	Esercizi di autovalutazione.....	591
	Risposte agli esercizi di autovalutazione	592
	Esercizi	592
CAPITOLO 11: GLI STREAM DI INPUT/OUTPUT DEL C++		595
11.1	Introduzione	595
11.2	Gli stream	596

	I file di intestazione della libreria iostream	597
	Le classi e gli oggetti che effettuano l'input/output su stream	597
11.3	L'output su stream	599
	L'operatore di inserimento nello stream	599
	Utilizzo degli operatori di inserimento/estrazione in cascata	601
	L'output delle variabili di tipo char*	602
	L'output di caratteri tramite put	602
11.4	L'input da stream	603
	L'operatore di estrazione dallo stream	603
	Le funzioni membro get e getline	606
	Altre funzioni membro di istream: peek, putback e ignore	608
	L'I/O type-safe	608
11.5	L'input/output non formattato delle funzioni read, gcount e write	609
11.6	I manipolatori di stream	609
	La base dei numeri interi su uno stream: dec, oct, hex e setbase	610
	La precisione dei valori a virgola mobile: precision e setprecision	610
	L'ampiezza dei campi: setw e width	612
	I manipolatori definiti dall'utente	614
11.7	I valori di stato della formattazione	615
	I flag di stato della formattazione	615
	Gli zero in coda e i punti decimali: ios::showpoint	616
	La giustificazione: ios::left, ios::right e ios::internal	617
	Il riempimento dei campi: fill e setfill	619
	La base dei numeri interi: ios::dec, ios::oct, ios::hex e ios::showbase	620
	I numeri a virgola mobile e la notazione scientifica: ios::scientific e ios::fixed	621
	Il controllo delle lettere maiuscole/minuscole: ios::uppercase	622
	L'attivazione e la disattivazione dei flag di formattazione: flags, setiosflags e resetiosflags	622
11.8	I valori di stato degli errori in uno stream	624
11.9	Il collegamento di uno stream di output a uno stream di input	626
	Esercizi di autovalutazione	626
	Risposte agli esercizi di autovalutazione	629
	Esercizi	631
	APPENDICE A: RIEPILOGO DEGLI OPERATORI	633
	APPENDICE B: L'INSIEME DEI CARATTERI ASCII	635
	APPENDICE C: I SISTEMI DI NUMERAZIONE	637
C.1	Introduzione	637
C.2	Usare i numeri ottali ed esadecimali per abbreviare i numeri binari	640
C.3	La conversione dei numeri ottali ed esadecimali in binari	642
C.4	La conversione da binario, ottale o esadecimale in decimale	642
C.5	La conversione da decimale a binario, ottale o esadecimale	643
C.6	I numeri binari negativi: la notazione in complemento a due	644

Esercizi di autovalutazione.....	646
Risposte agli esercizi di autovalutazione	647
Esercizi	648
APPENDICE D: IL C++ SU INTERNET	649
D.1 Risorse	649
D.2 Tutorial	649
D.3 FAQ (risposte alle domande più comuni)	650
D.4 comp.lang.c++	650
D.5 Compilatori	651
D.6 Standard Template Library	652
Tutorial	652
Riferimenti	652
FAQ	652
Articoli, libri e interviste	653
Software	653
INDICE ANALITICO	655