

List of Figures

FIGURE 1-1: ISOMORPHIC MAPPING BETWEEN PROBLEM DOMAIN AND DESIGN DOMAIN	9
FIGURE 2-1: MICROSOFT.NET FRAMEWORK INSTALLATION DIRECTORY	21
FIGURE 2-2: PARTIAL DIRECTORY LISTING OF THE v3.5 FOLDER	21
FIGURE 2-3: CREATING AN ENVIRONMENT VARIABLE	23
FIGURE 2-4: EDITING THE PATH USER ENVIRONMENT VARIABLE	24
FIGURE 2-5: COMMAND CONSOLE WINDOW	24
FIGURE 2-6: TESTING THE DOT_NET_FRAMEWORK_HOME ENVIRONMENT VARIABLE	24
FIGURE 2-7: TESTING THE PATH ENVIRONMENT VARIABLE BY RUNNING THE C# COMPILER	25
FIGURE 2-8: CREATING A NEW FOLDER	25
FIGURE 2-9: PROJECTS FOLDER BEFORE SETTING FOLDER OPTIONS	26
FIGURE 2-10: FOLDER OPTIONS DIALOG WINDOW	26
FIGURE 2-11: PROJECTS FOLDER AFTER SETTING FOLDER OPTIONS	27
FIGURE 2-12: DEFAULT COMMAND CONSOLE WINDOW	27
FIGURE 2-13: COMMAND CONSOLE PROPERTIES DIALOG	28
FIGURE 2-14: SETTING THE START IN PROPERTY	28
FIGURE 2-15: SETTING COMMAND CONSOLE LAYOUT PROPERTIES	29
FIGURE 2-16: DIRECTORY LISTING OF THE CHAPTER2 DIRECTORY SHOWING THE HelloWorld.cs FILE	30
FIGURE 2-17: COMPILING HelloWorld.cs USING THE CSC C# COMPILER COMMAND	30
FIGURE 2-18: RUNNING THE HelloWorld PROGRAM	30
FIGURE 2-19: COMPILER OUTPUT SHOWING COMPILER ERROR ON LINE 6 AT POSITION 39	31
FIGURE 2-20: C# LANGUAGE COMPILER ERRORS	31
FIGURE 2-21: C# COMPILER ERROR CS1002 “; EXPECTED”	32
FIGURE 2-22: VISUAL C# EXPRESS INSTALLATION WINDOW	33
FIGURE 2-23: VISUAL C# EXPRESS INITIAL START-UP SCREEN	34
FIGURE 2-24: NEW PROJECT DIALOG SHOWING CONSOLE APPLICATION SELECTED	34
FIGURE 2-25: HelloWorld PROJECT VIEW	35
FIGURE 2-26: INTELLISENSE POP-UP WINDOW SHOWING AVAILABLE CONSOLE OBJECT METHODS AND PROPERTIES	35
FIGURE 2-27: UPDATED HelloWorld VISUAL C# PROJECT	36
FIGURE 2-28: SAVING THE HelloWorld PROJECT	36
FIGURE 2-29: BUILDING HelloWorld PROJECT	37
FIGURE 2-30: RESULTS OF RUNNING THE tree /f COMMAND FROM THE COMMAND PROMPT	37
FIGURE 3-1: TIGHT-SPIRAL DEVELOPMENT CYCLE DEPLOYMENT	43
FIGURE 3-2: ROBOT RAT VIEWED AS A COLLECTION OF ATTRIBUTES	47
FIGURE 3-3: ROBOT RAT FLOOR SKETCH	47
FIGURE 3-4: COMPLETE ROBOT RAT ATTRIBUTES	48
FIGURE 3-5: ROBOTRAT UML CLASS DIAGRAM	50
FIGURE 3-6: COMPILING AND TESTING ROBOTRAT – FIRST ITERATION	52
FIGURE 3-7: COMPILING & TESTING ROBOTRAT - SECOND ITERATION	54
FIGURE 3-8: TESTING MENU COMMANDS	57
FIGURE 3-9: A DISTURBING ERROR MESSAGE	58
FIGURE 3-10: UNHANDLED INDEXOUTOFRangeException ERROR MESSAGE	58
FIGURE 3-11: pen_position STATE TRANSITION DIAGRAM	60
FIGURE 3-12: STATE TRANSITION DIAGRAM FOR THE direction VARIABLE	60
FIGURE 3-13: TESTING THE PrintFloor() METHOD	62
FIGURE 3-14: TESTING ROBOT RAT MOVEMENT IN ALL DIRECTIONS	66
FIGURE 3-15: ROBOT RAT HTML DOCUMENTATION GENERATED WITH DOXYGEN	72
FIGURE 4-1: TYPICAL APPLE MAC PRO COMPUTER SYSTEM	76
FIGURE 4-2: SYSTEM UNIT COMPONENTS	77
FIGURE 4-3: MAIN LOGIC BOARD BLOCK DIAGRAM	77
FIGURE 4-4: INTEL XEON 5100 DUAL CORE PROCESSOR	78

FIGURE 4-5: INTEL XEON 5100 DUAL-CORE MICROPROCESSOR BLOCK DIAGRAMS	78
FIGURE 4-6: MEMORY HIERARCHY	80
FIGURE 4-7: SIMPLIFIED MEMORY SUBSYSTEM DIAGRAM	80
FIGURE 4-8: SIMPLIFIED MAIN MEMORY DIAGRAM	81
FIGURE 4-9: PROCESSING CYCLE	83
FIGURE 4-10: DUMB SORT RESULTS 1	85
FIGURE 4-11: DUMB SORT RESULTS 2	85
FIGURE 4-12: DUMB SORT RESULTS 3	85
FIGURE 4-13: ALGORITHMIC GROWTH RATES	85
FIGURE 4-14: THE C# COMPILE AND EXECUTION PROCESS OVERVIEW	86
FIGURE 4-15: MSIL DISASSEMBLER SESSION SHOWING MAIN() METHOD IL INSTRUCTIONS	87
FIGURE 4-16: THE COMMON LANGUAGE INFRASTRUCTURE ARCHITECTURE	88
FIGURE 4-17: MANAGED ASSEMBLIES CAN BE EXECUTED ON ANY SYSTEM THAT IMPLEMENTS THE COMMON LANGUAGE INFRASTRUCTURE	89
FIGURE 4-18: CHAPTER 3'S ROBOT RAT PROGRAM RUNNING IN THE MONO ENVIRONMENT ON APPLE OS X	89
FIGURE 4-19: MICROSOFT .NET ARCHITECTURE	90
FIGURE 5-1: .NET FRAMEWORK CLASS LIBRARY REFERENCE PAGE	94
FIGURE 5-2: .NET DEVELOPMENT LINK EXPANDED AND CLASS LIBRARY LINK HIGHLIGHTED	95
FIGURE 5-3: CLASS LIBRARY LINK EXPANDED AND SYSTEM NAMESPACE HIGHLIGHTED	95
FIGURE 5-4: STRING CLASS API REFERENCE OVERVIEW PAGE	96
FIGURE 5-5: STRING MEMBERS PAGE	97
FIGURE 5-6: STRING CLASS'S PUBLIC CONSTRUCTORS PARTIAL LISTING	98
FIGURE 5-7: STRING CLASS'S METHODS PAGE PARTIAL LISTING	98
FIGURE 5-8: STRING.SUBSTRING METHOD PAGE	99
FIGURE 5-9: STRING.SUBSTRING PAGE WITH COLLAPSED SUBHEADINGS	99
FIGURE 5-10: STRING.SUBSTRING EXAMPLE SECTION EXPANDED SHOWING EXAMPLE CODE	99
FIGURE 5-11: STRING CLASS INHERITANCE HIERARCHY	101
FIGURE 5-12: OBSOLETE .NET FRAMEWORK VERSION 2.0 API PARTIAL LISTING BY NAMESPACE	102
FIGURE 6-1: RESULTS OF RUNNING EXAMPLE 6.1	112
FIGURE 6-2: RESULTS OF COMPILING EXAMPLE 6.3 WITH IMPROPER MAIN() METHOD SIGNATURE	113
FIGURE 6-3: ERRORS PRODUCED WHEN ATTEMPTING TO REINTRODUCE A RESERVED KEYWORD	114
FIGURE 6-4: C# TYPE HIERARCHY	115
FIGURE 6-5: RESULTS OF RUNNING EXAMPLE 6.6	116
FIGURE 6-6: THE RESULTS OF RUNNING EXAMPLE 6.7	117
FIGURE 6-7: VALUE TYPE MEMORY ALLOCATION	117
FIGURE 6-8: REFERENCE TYPE MEMORY ALLOCATION	117
FIGURE 6-9: RESULTS OF CALLING THE APPEND() METHOD VIA THE sb1 VARIABLE	117
FIGURE 6-10: RESULTS OF RUNNING EXAMPLE 6.9	122
FIGURE 6-11: RESULTS OF RUNNING EXAMPLE 6.10	123
FIGURE 6-12: RESULTS OF RUNNING EXAMPLE 6.11	124
FIGURE 6-13: RESULTS OF RUNNING EXAMPLE 6.12	124
FIGURE 6-14: RESULTS OF RUNNING EXAMPLE 6.13	126
FIGURE 6-15: LOGICAL AND, OR, AND XOR TRUTH TABLES	127
FIGURE 6-16: RESULTS OF RUNNING EXAMPLE 6.14	127
FIGURE 6-17: RESULTS OF RUNNING EXAMPLE 6.15	128
FIGURE 6-18: RESULTS OF RUNNING EXAMPLE 6.16	129
FIGURE 6-19: RESULTS OF RUNNING EXAMPLE 6.17	129
FIGURE 6-20: COMPILER WARNING DUE TO UNREACHABLE CODE	130
FIGURE 6-21: RESULTS OF RUNNING EXAMPLE 6.18	130
FIGURE 6-22: RESULTS OF RUNNING EXAMPLE 6.19	131
FIGURE 7-1: IF STATEMENT EXECUTION DIAGRAM	136
FIGURE 7-2: RESULTS OF RUNNING EXAMPLE 7.1	137
FIGURE 7-3: TYPICAL .NET ERROR MESSAGE DIALOG WINDOW	137
FIGURE 7-4: UNHANDLED INDEXOUTOFRangeException MESSAGE	137
FIGURE 7-5: FORMATException ERROR MESSAGE	138
FIGURE 7-6: RESULTS OF RUNNING EXAMPLE 7.2	138
FIGURE 7-7: RESULTS OF RUNNING EXAMPLE 7.2	139
FIGURE 7-8: RESULTS OF RUNNING EXAMPLE 7.3	140
FIGURE 7-9: IF/ELSE STATEMENT EXECUTION DIAGRAM	140
FIGURE 7-10: RESULTS OF RUNNING EXAMPLE 7.5	141

FIGURE 7-11: RESULTS OF RUNNING EXAMPLE 7.6	142
FIGURE 7-12: SWITCH STATEMENT EXECUTION DIAGRAM	142
FIGURE 7-13: RESULTS OF RUNNING EXAMPLE 7.7	143
FIGURE 7-14: RESULTS OF RUNNING EXAMPLE 7.8	144
FIGURE 7-15: RESULTS OF RUNNING EXAMPLE 7.9	145
FIGURE 7-16: WHILE STATEMENT EXECUTION DIAGRAM	146
FIGURE 7-17: RESULTS OF RUNNING EXAMPLE 7.10	146
FIGURE 7-18: DO/WHILE STATEMENT EXECUTION DIAGRAM	147
FIGURE 7-19: RESULTS OF RUNNING EXAMPLE 7.11	147
FIGURE 7-20: FOR STATEMENT EXECUTION DIAGRAM	148
FIGURE 7-21: RESULTS OF RUNNING EXAMPLE 7.12	149
FIGURE 7-22: RESULTS OF RUNNING EXAMPLE 7.13	150
FIGURE 7-23: RESULTS OF RUNNING CHECKBOOKBALANCER	151
FIGURE 7-24: RESULTS OF RUNNING EXAMPLE 7.15	152
FIGURE 7-25: RESULTS OF RUNNING EXAMPLE 7.16	153
FIGURE 7-26: RESULTS OF RUNNING EXAMPLE 7.17	153
FIGURE 8-1: ARRAY ELEMENTS ARE CONTIGUOUS AND HOMOGENEOUS	162
FIGURE 8-2: DECLARING A SINGLE-DIMENSIONAL ARRAY	163
FIGURE 8-3: ARRAY-TYPE INHERITANCE HIERARCHY	164
FIGURE 8-4: RESULTS OF RUNNING EXAMPLE 8.1	166
FIGURE 8-5: MEMORY REPRESENTATION OF VALUE TYPE ARRAY <code>int_array</code> SHOWING DEFAULT INITIALIZATION	167
FIGURE 8-6: RESULTS OF RUNNING EXAMPLE 8.2	167
FIGURE 8-7: ELEMENT VALUES OF <code>int_array</code> AFTER INITIALIZATION PERFORMED BY SECOND <code>for</code> LOOP	168
FIGURE 8-8: RESULTS OF RUNNING EXAMPLE 8.3	168
FIGURE 8-9: RESULTS OF RUNNING EXAMPLE 8.4	169
FIGURE 8-10: RESULTS OF RUNNING EXAMPLE 8.5	170
FIGURE 8-11: STATE OF AFFAIRS AFTER LINE 5 OF EXAMPLE 8.5 EXECUTES	170
FIGURE 8-12: STATE OF AFFAIRS AFTER LINE 10 OF EXAMPLE 8.5 EXECUTES	171
FIGURE 8-13: STATE OF AFFAIRS AFTER LINE 14 OF EXAMPLE 8.5 EXECUTES	171
FIGURE 8-14: FINAL STATE OF AFFAIRS: ALL <code>object_array</code> ELEMENTS POINT TO AN OBJECT OBJECT	172
FIGURE 8-15: RESULTS OF RUNNING EXAMPLE 8.6	173
FIGURE 8-16: RESULTS OF RUNNING EXAMPLE 8.7	174
FIGURE 8-17: RESULTS OF RUNNING EXAMPLE 8.8	175
FIGURE 8-18: RECTANGULAR ARRAY DECLARATION SYNTAX	176
FIGURE 8-19: ACCESSING TWO-DIMENSIONAL ARRAY ELEMENTS	177
FIGURE 8-20: RESULTS OF RUNNING EXAMPLE 8.9	177
FIGURE 8-21: RESULTS OF RUNNING EXAMPLE 8.10	178
FIGURE 8-22: ARRAY DECLARATION SYNTAX FOR A TWO-DIMENSIONAL RAGGED ARRAY	179
FIGURE 8-23: RESULTS OF RUNNING EXAMPLE 8.11	179
FIGURE 8-24: RESULTS OF RUNNING EXAMPLE 8.12	181
FIGURE 8-25: RESULTS OF RUNNING EXAMPLE 8.13	182
FIGURE 8-26: RESULTS OF RUNNING EXAMPLE 8.14	183
FIGURE 9-1: PEOPLE MANAGEMENT PROGRAM PROJECT SPECIFICATION	191
FIGURE 9-2: CLASS DIAGRAM FOR PEOPLE MANAGER CLASSES	193
FIGURE 9-3: STATIC AND NON-STATIC FIELDS	195
FIGURE 9-4: RESULTS OF RUNNING EXAMPLE 9.1	196
FIGURE 9-5: ERROR RESULTING FROM AN ATTEMPT TO ASSIGN TO A READONLY FIELD	196
FIGURE 9-6: RESULTS OF RUNNING EXAMPLE 9.3	197
FIGURE 9-7: RESULTS OF RUNNING EXAMPLE 9.4	197
FIGURE 9-8: RESULTS OF RUNNING EXAMPLE 9.5	199
FIGURE 9-9: HORIZONTAL ACCESS CONTROLLED VIA ACCESS MODIFIERS <code>public</code> AND <code>private</code>	202
FIGURE 9-10: METHOD DEFINITION STRUCTURE	203
FIGURE 9-11: RESULTS OF RUNNING EXAMPLE 9.10	210
FIGURE 9-12: RESULTS OF RUNNING EXAMPLE 9.12	211
FIGURE 9-13: RESULTS OF RUNNING EXAMPLE 9.14	213
FIGURE 9-14: RESULTS OF RUNNING EXAMPLE 9.16	214
FIGURE 9-15: RESULTS OF RUNNING EXAMPLE 9.21	217
FIGURE 9-16: RESULTS OF RUNNING EXAMPLE 9.23	219
FIGURE 9-17: DEFAULT VALUE PARAMETER BEHAVIOR	220

FIGURE 9-18: REFERENCE PARAMETER BEHAVIOR – Using ref Modifier	221
FIGURE 9-19: RESULTS OF RUNNING EXAMPLE 9.24	222
FIGURE 9-20: RESULTS OF RUNNING EXAMPLE 9.25	222
FIGURE 9-21: RESULTS OF RUNNING EXAMPLE 9.26	223
FIGURE 9-22: RESULTS OF RUNNING EXAMPLE 9.27	224
FIGURE 9-23: STRUCTURES VS. VALUE TYPES	225
FIGURE 9-24: RESULTS OF RUNNING EXAMPLE 9.28	227
FIGURE 9-25: CIRCULAR LINKED LIST WITH THREE NODES	230
FIGURE 10-1: UML DIAGRAM SHOWING SIMPLE AGGREGATION	237
FIGURE 10-2: PART CLASS SHARED BETWEEN SIMPLE AGGREGATE CLASSES	237
FIGURE 10-3: UML DIAGRAM SHOWING COMPOSITE AGGREGATION	237
FIGURE 10-4: SIMPLE AGGREGATION EXAMPLE	238
FIGURE 10-5: RESULTS OF RUNNING EXAMPLE 10.3	239
FIGURE 10-6: COMPOSITE AGGREGATION EXAMPLE	239
FIGURE 10-7: RESULTS OF RUNNING EXAMPLE 10.6	240
FIGURE 10-8: SEQUENCE DIAGRAM – SIMPLE AGGREGATION	240
FIGURE 10-9: SEQUENCE DIAGRAM – COMPOSITE AGGREGATION	241
FIGURE 10-10: ENGINE SIMULATION PROJECT SPECIFICATION	242
FIGURE 10-11: ENGINE SIMULATION CLASS DIAGRAM	243
FIGURE 10-12: ENGINE CLASS DIAGRAM	243
FIGURE 10-13: CREATE ENGINE OBJECT SEQUENCE	245
FIGURE 10-14: RESULT OF RUNNING EXAMPLE 10.7	246
FIGURE 10-15: SIMPLE AGGREGATION CLASS DIAGRAM	251
FIGURE 10-16: COMPOSITE AGGREGATION CLASS DIAGRAM	251
FIGURE 11-1: INHERITANCE HIERARCHY ILLUSTRATING GENERALIZED AND SPECIALIZED BEHAVIOR	256
FIGURE 11-2: UML CLASS DIAGRAM SHOWING DERIVEDCLASS INHERITING FROM BASECLASS	258
FIGURE 11-3: UML DIAGRAM OF BASECLASS AND DERIVEDCLASS SHOWING FIELDS, PROPERTIES, AND METHODS	259
FIGURE 11-4: RESULTS OF RUNNING EXAMPLE 11.3	261
FIGURE 11-5: UML DIAGRAM SHOWING STUDENT CLASS INHERITANCE HIERARCHY	262
FIGURE 11-6: RESULTS OF RUNNING EXAMPLE 11.6	264
FIGURE 11-7: RESULTS OF RUNNING EXAMPLE 11.7	265
FIGURE 11-8: UML CLASS DIAGRAM FOR BASECLASS & DERIVEDCLASS	266
FIGURE 11-9: RESULTS OF RUNNING EXAMPLE 11.3 WITH MODIFIED VERSIONS OF BASECLASS AND DERIVEDCLASS	267
FIGURE 11-10: EXPRESSING AN ABSTRACT CLASS IN THE UML	268
FIGURE 11-11: UML CLASS DIAGRAM SHOWING THE ABSTRACTCLASS AND DERIVEDCLASS INHERITANCE HIERARCHY	269
FIGURE 11-12: RESULTS OF RUNNING EXAMPLE 11.12	270
FIGURE 11-13: TWO TYPES OF UML INTERFACE DIAGRAMS	271
FIGURE 11-14: UML DIAGRAM SHOWING THE SIMPLE FORM OF REALIZATION	272
FIGURE 11-15: UML DIAGRAM SHOWING THE EXPANDED FORM OF REALIZATION	272
FIGURE 11-16: UML DIAGRAM SHOWING THE MESSAGEPRINTER CLASS IMPLEMENTING THE IMessagePrinter INTERFACE	272
FIGURE 11-17: RESULTS OF RUNNING EXAMPLE 11.15	273
FIGURE 11-18: EMPLOYEE CLASS INHERITANCE HIERARCHY	276
FIGURE 11-19: RESULTS OF RUNNING EXAMPLE 11.20	278
FIGURE 11-20: ENGINE SIMULATION UML CLASS DIAGRAM	279
FIGURE 11-21: RESULTS OF RUNNING THE ENGINETESTAPP	280
FIGURE 12-1: FORM CLASS INHERITANCE HIERARCHY	292
FIGURE 12-2: RESULTS OF RUNNING EXAMPLE 12.1	293
FIGURE 12-3: A STANDARD WINDOW CAN BE RESIZED BY DRAGGING THE LOWER RIGHT CORNER	294
FIGURE 12-4: WINDOWS MESSAGE ROUTING (MESSAGE PUMP)	295
FIGURE 12-5: RESULTS OF RUNNING EXAMPLE 12.2	296
FIGURE 12-6: SCREEN COORDINATE SYSTEM	297
FIGURE 12-7: WINDOW COORDINATES	298
FIGURE 12-8: RESULTS OF RUNNING EXAMPLE 12.3	299
FIGURE 12-9: RUNNING EXAMPLE 12.4 VIA THE COMMAND LINE WITH THE NAME OF THE IMAGE WCC_2.jpg	300
FIGURE 12-10: RUNNING EXAMPLE 12.4 WITH NO IMAGE	301
FIGURE 12-11: RESULTS OF RUNNING EXAMPLE 12.5	302
FIGURE 12-12: RESULTS OF RUNNING EXAMPLE 12.6 WITH DIFFERENT TEXT IN THE TEXTBOX	305
FIGURE 12-13: UML CLASS DIAGRAM SHOWING SEPARATE GUI AND APPLICATION/EVENT HANDLER CLASSES	306
FIGURE 12-14: RESULTS OF RUNNING EXAMPLE 12.8 – GUI EVENTS HANDLED IN SEPARATE OBJECT	308

FIGURE 12-15: RESULTS OF RUNNING EXAMPLE 12.10 – BUTTONS ADJUST WHEN WINDOW IS RESIZED	310
FIGURE 12-16: RESULTS OF RUNNING EXAMPLE 12.12 AFTER SEVERAL BUTTONS HAVE BEEN CLICKED	311
FIGURE 12-17: WINDOW AND MENU STRUCTURE OF MENU DEMO PROGRAM	312
FIGURE 12-18: RESULTS OF RUNNING EXAMPLE 12.14 AND ADDING SEVERAL BUTTONS AND TEXT BOXES	314
FIGURE 12-19: RESULTS OF RUNNING EXAMPLE 12.16 – DOUBLE-CLICKING THE FIRST LINE	316
FIGURE 13-1: EVENT PUBLISHER AND SUBSCRIBER	322
FIGURE 13-2: EVENT PUBLISHER AND SUBSCRIBER	323
FIGURE 13-3: MINUTE TICK UML CLASS DIAGRAM	324
FIGURE 13-4: RESULTS OF RUNNING EXAMPLE 13.5	325
FIGURE 13-5: WATER TANK SYSTEM UML CLASS DIAGRAM	326
FIGURE 13-6: RESULTS OF RUNNING EXAMPLE 13.11	332
FIGURE 14-1: RESULTS OF TESTING DYNAMICARRAY	340
FIGURE 14-2: RESULTS OF RUNNING EXAMPLE 17.3	341
FIGURE 14-3: RESULTS OF RUNNING EXAMPLE 14.4	341
FIGURE 14-4: ARRAY OF OBJECT REFERENCES BEFORE INSERTION	342
FIGURE 14-5: NEW REFERENCE TO BE INSERTED AT ARRAY ELEMENT 3 (INDEX 2)	342
FIGURE 14-6: ARRAY AFTER NEW REFERENCE INSERTION	343
FIGURE 14-7: LINKED LIST NODE ORGANIZATION	343
FIGURE 14-8: LINKED LIST BEFORE NEW ELEMENT INSERTION	344
FIGURE 14-9: NEW REFERENCE BEING INSERTED INTO SECOND ELEMENT POSITION	344
FIGURE 14-10: REFERENCES OF PREVIOUS, NEW, AND NEXT LIST ELEMENTS MUST BE MANIPULATED	344
FIGURE 14-11: LINKED LIST INSERTION COMPLETE	345
FIGURE 14-12: A HASH FUNCTION TRANSFORMS A KEY VALUE INTO AN ARRAY INDEX	345
FIGURE 14-13: HASH TABLE COLLISIONS ARE RESOLVED BY LINKING NODES TOGETHER	346
FIGURE 14-14: RED-BLACK TREE NODE DATA ELEMENTS	346
FIGURE 14-15: RED-BLACK TREE AFTER INSERTING INTEGER VALUES 9, 3, 5, 6, 7, 8, 4, 1	346
FIGURE 14-16: A STACK AFTER SEVERAL PUSH AND POP OPERATIONS	347
FIGURE 14-17: A QUEUE AFTER SEVERAL ENQUEUE AND DEQUEUE OPERATIONS	347
FIGURE 14-18: RESULTS OF RUNNING EXAMPLE 14.6	352
FIGURE 14-19: RESULTS OF RUNNING EXAMPLE 14.8	353
FIGURE 14-20: RESULTS OF RUNNING EXAMPLE 14.9	355
FIGURE 14-21: RESULTS OF RUNNING EXAMPLE 14.11	356
FIGURE 14-22: RESULTS OF RUNNING EXAMPLE 14.13	359
FIGURE 14-23: RESULTS OF RUNNING EXAMPLE 14.15	360
FIGURE 14-24: RESULTS OF RUNNING EXAMPLE 14.16	361
FIGURE 15-1: EXCEPTION INFORMATION TABLE	367
FIGURE 15-2: EXCEPTION CLASS HIERARCHY	368
FIGURE 15-3: GETTING EXCEPTION INFORMATION FROM MSDN	369
FIGURE 15-4: RESULTS OF RUNNING EXAMPLE 15.1	371
FIGURE 15-5: RESULTS OF RUNNING EXAMPLE 15.2	372
FIGURE 15-6: RESULTS OF RUNNING EXAMPLE 15.3	373
FIGURE 15-7: RESULTS OF RUNNING EXAMPLE 15.4	373
FIGURE 15-8: RESULTS OF RUNNING EXAMPLE 15.7	376
FIGURE 16-1: LIST OF RUNNING APPLICATIONS	383
FIGURE 16-2: PARTIAL LIST OF PROCESSES RUNNING ON THE SAME COMPUTER	383
FIGURE 16-3: PROCESSES AND THEIR THREADS EXECUTING IN A SINGLE-PROCESSOR ENVIRONMENT	384
FIGURE 16-4: PROCESSES AND THEIR THREADS EXECUTING IN A MULTIPROCESSOR ENVIRONMENT	385
FIGURE 16-5: SINGLETHREADEDVACATION PROGRAM OUTPUT	387
FIGURE 16-6: MULTITHREADEDVACATION PROGRAM OUTPUT - PARTIAL LISTING	388
FIGURE 16-7: THREAD STATES AND TRANSITION INITIATORS	389
FIGURE 16-8: RESULTS OF RUNNING EXAMPLE 16.3	390
FIGURE 16-9: RESULTS OF RUNNING EXAMPLE 16.4	391
FIGURE 16-10: RESULTS OF RUNNING EXAMPLE 16.5	392
FIGURE 16-11: RESULTS OF RUNNING EXAMPLE 16.6	393
FIGURE 16-12: RESULTS OF RUNNING EXAMPLE 16.7	393
FIGURE 16-13: RESULTS OF RUNNING EXAMPLE 16.8	394
FIGURE 16-14: RESULTS OF RUNNING EXAMPLE 16.9	395
FIGURE 16-15: ONE PARTICULAR RESULT OF RUNNING EXAMPLE 16.10	398
FIGURE 16-16: PARTIAL RESULT OF RUNNING EXAMPLE 16.11	400

FIGURE 16-17: RESULTS OF RUNNING EXAMPLE 16.12	401
FIGURE 16-18: RESULTS OF RUNNING EXAMPLE 16.13	402
FIGURE 16-19: RESULTS OF RUNNING EXAMPLE 16.14	403
FIGURE 17-1: SIMPLIFIED VIEW OF SERVICE LAYERS	410
FIGURE 17-2: TYPICAL DIRECTORY STRUCTURE	411
FIGURE 17-3: THE ABSOLUTE PATH TO THE REPORTS\EAST\Q2.xls FILE	411
FIGURE 17-4: RESULTS OF RUNNING EXAMPLE 17.1	412
FIGURE 17-5: RESULTS OF RUNNING EXAMPLE 17.3	416
FIGURE 17-6: RESULTS OF RUNNING EXAMPLE 17.4	417
FIGURE 17-7: RESULTS OF RUNNING EXAMPLE 17.6	420
FIGURE 17-8: RESULTS OF RUNNING EXAMPLE 17.8	422
FIGURE 17-9: LEGACY DATAFILE ADAPTER PROJECT SPECIFICATION	423
FIGURE 17-10: HEADER AND RECORD LENGTH ANALYSIS	424
FIGURE 17-11: MONITOR.ENTER()/MONITOR.EXIT() VS. THE LOCK KEYWORD	425
FIGURE 17-12: RESULTS OF RUNNING EXAMPLE 17.16 ONCE	437
FIGURE 17-13: RESULTS OF RUNNING EXAMPLE 17.19	440
FIGURE 17-14: RESULTS OF RUNNING EXAMPLE 17.21 AND SELECTING THREE FILES	442
FIGURE 18-1: A SIMPLE COMPUTER NETWORK	450
FIGURE 18-2: LOCAL AREA NETWORK CONNECTED TO THE INTERNET	451
FIGURE 18-3: THE INTERNET – A NETWORK OF NETWORKS COMMUNICATING VIA INTERNET PROTOCOLS	452
FIGURE 18-4: CLIENT AND SERVER HARDWARE AND APPLICATIONS	453
FIGURE 18-5: CLIENT AND SERVER APPLICATIONS PHYSICALLY DEPLOYED TO THE SAME COMPUTER	454
FIGURE 18-6: RUNNING MULTIPLE CLIENTS ON SAME HARDWARE	455
FIGURE 18-7: CLIENT AND SERVER APPLICATIONS DEPLOYED ON DIFFERENT COMPUTERS	455
FIGURE 18-8: A MULTITIERED APPLICATION	456
FIGURE 18-9: PHYSICALLY DEPLOYING LOGICAL APPLICATION TIERS ON SAME COMPUTER	457
FIGURE 18-10: LOGICAL APPLICATION TIERS PHYSICALLY DEPLOYED TO DIFFERENT COMPUTERS	457
FIGURE 18-11: TCP/IP PROTOCOL STACK	458
FIGURE 18-12: INTERNET PROTOCOL STACK OPERATIONS	459
FIGURE 19-1: .NET REMOTING ARCHITECTURE	466
FIGURE 19-2: REMOTINGSERVER WAITING FOR SOMETHING TO DO	468
FIGURE 19-3: RESULTS OF RUNNING REMOTINGSERVER AND REMOTINGCLIENT WITH A SINGLECALL MODE REMOTE OBJECT	469
FIGURE 19-4: RESULTS OF HOSTING TESTCLASS REMOTE OBJECT IN SINGLETON MODE	470
FIGURE 19-5: RESULTS OF ACCESSING A REMOTE OBJECT VIA AN INTERFACE	472
FIGURE 19-6: RESULTS OF RUNNING REMOTINGSERVER AND REMOTINGCLIENT WITH CONFIGURATION FILES	473
FIGURE 19-7: RESULTS OF SENDING A COLLECTION OF PERSON OBJECTS TO A REMOTING CLIENT	477
FIGURE 19-8: SERVER APPLICATION LISTENS ON A HOST AND PORT FOR INCOMING TCPCLIENT CONNECTIONS	478
FIGURE 19-9: TCPListener ACCEPTS INCOMING TCPCLIENT CONNECTION	478
FIGURE 19-10: TCPCLIENTS COMMUNICATE VIA A NETWORKSTREAM USING STREAMREADER AND STREAMWRITER OBJECTS	479
FIGURE 19-11: RESULTS OF RUNNING THE ECHOCLIENT AND ECHOSERVER APPLICATIONS	481
FIGURE 19-12: TWO CLIENTS CONNECTED TO MULTITHREADEDCLIENTSERVER	482
FIGURE 19-13: RESULTS OF RUNNING MULTIPLECHOSEVER AND ECHOCLIENT (MOD 1) APPLICATIONS	485
FIGURE 19-14: RESULTS OF RUNNING SURREALISTECHOSEVER AND ECHOCLIENT (MOD 2)	489
FIGURE 20-1: EMPLOYEE TRAINING SERVER APPLICATION ARCHITECTURE	494
FIGURE 20-2: SQL SERVER SYSTEM CONFIGURATION CHECK	496
FIGURE 20-3: SQL EXPRESS FEATURE SELECTION DIALOG	496
FIGURE 20-4: RESULTS OF TESTING SQL SERVER EXPRESS EDITION INSTALLATION	497
FIGURE 20-5: MANAGEMENT STUDIO LOGIN DIALOG	497
FIGURE 20-6: SQL MANAGEMENT STUDIO MAIN WINDOW	497
FIGURE 20-7: ENTERPRISE LIBRARY CUSTOM SETUP DIALOG	498
FIGURE 20-8: DOUBLE-CLICK THE INSTALLSERVICES.BAT FILE	498
FIGURE 20-9: ENTERPRISE LIBRARY CONFIGURATION FILE CREATION TOOL	500
FIGURE 20-10: CONTENTS OF THE SIMPLECONNECTION PROJECT DIRECTORY BEFORE COMPILING	500
FIGURE 20-11: RESULTS OF RUNNING THE SIMPLECONNECTION APPLICATION	500
FIGURE 20-12: THE PRIMARY KEY OF ONE TABLE CAN SERVE AS THE FOREIGN KEY IN A RELATED TABLE	501
FIGURE 20-13: SQL SERVER'S DEFAULT DATABASES	502
FIGURE 20-14: CREATING EMPLOYEETRAINING DATABASE WITH SQL COMMAND UTILITY	503
FIGURE 20-15: CHECKING ON THE EXISTENCE OF THE EMPLOYEETRAINING DATABASE	503
FIGURE 20-16: RESULTS OF EXECUTING THE CREATE_DATABASE.SQL SCRIPT	504

FIGURE 20-17: RESULTS OF EXECUTING CREATE_TABLES.SQL DATABASE SCRIPT	504
FIGURE 20-18: RESULTS OF RUNNING CREATE_TEST_DATA.SQL DATABASE SCRIPT	507
FIGURE 20-19: RESULTS OF EXECUTING A SIMPLE SELECT STATEMENT	507
FIGURE 20-20: SELECTING SPECIFIC ROWS WITH SELECT STATEMENT	508
FIGURE 20-21: INSERTING MORE TEST DATA WITH THE CREATE_TEST_DATA.SQL DATABASE SCRIPT	508
FIGURE 20-22: RESULTS OF LIMITING DATA RETURNED FROM SELECT STATEMENT WITH WHERE CLAUSE	509
FIGURE 20-23: RESULTS OF EXECUTING THE PREVIOUS QUERY	509
FIGURE 20-24: CHANGING CORALIE POWELL'S LAST NAME TO MILLER WITH THE UPDATE STATEMENT	510
FIGURE 20-25: DELETING ALL EMPLOYEES WHOSE LAST NAMES = "MILLER"	510
FIGURE 20-26: VERIFYING THE CREATION OF THE tbl_EMPLOYEE_TRAINING TABLE	512
FIGURE 20-27: SELECTING EMPLOYEEIDS FROM tbl_EMPLOYEE	512
FIGURE 20-28: RESULTS OF RUNNING THE PREVIOUS SQL QUERY	514
FIGURE 20-29: RESULTS OF RUNNING THE PREVIOUS SQL QUERY	515
FIGURE 20-30: RESULTS OF EXECUTING A CASCADE DELETE AND CHECKING THE RESULTS	515
FIGURE 20-31: EMPLOYEE TRAINING PROJECT FOLDER ARRANGEMENT	516
FIGURE 20-32: EMPLOYEEVO AND EMPLOYEEDAO CLASS DIAGRAM	520
FIGURE 20-33: RESULTS OF RUNNING THE COMPILEVO TARGET USING THE MSBUILD UTILITY	522
FIGURE 20-34: BUILD WARNINGS FROM CONFLICTING TYPE DECLARATIONS	522
FIGURE 20-35: INITIAL STATE OF THE EMPLOYEETRAININGSERVER APPLICATION WINDOW	530
FIGURE 20-36: EMPLOYEE PICTURE LOADED AND CREATE BUTTON ENABLED	531
FIGURE 20-37: TESTING WITH MORE EMPLOYEE PICTURES	531
FIGURE 20-38: TESTING THE INSERTION AND RETRIEVAL OF A LARGE IMAGE	531
FIGURE 20-39: TRAININGDAO AND TRAININGVO CLASS DIAGRAM	532
FIGURE 20-40: EMPLOYEEADMINBO UML CLASS DIAGRAM	533
FIGURE 20-41: COLLAPSED CODE REGIONS IN NOTEPAD++	541
FIGURE 20-42: MODIFIED TEST APPLICATION	543
FIGURE 20-43: EMPLOYEETRAININGREMOTEOBJECT UML CLASS DIAGRAM	552
FIGURE 20-44: EMPLOYEETRAININGSERVER RUNNING AND READY FOR REMOTE CONNECTIONS	555
FIGURE 20-45: CLIENT PROJECT DIRECTORY STRUCTURE	556
FIGURE 20-46: RUNNING CLIENT APPLICATION VIA THE MSBUILD PROJECT'S RUN TARGET	558
FIGURE 20-47: EMPLOYEETRAININGCLIENT UML CLASS DIAGRAM	559
FIGURE 20-48: MOCK-UP SKETCH OF THE EMPLOYEETRAININGAPPLICATION GUI	559
FIGURE 20-49: EMPLOYEETRAININGCLIENT INITIAL DISPLAY ON STARTUP – SOMETHING'S NOT QUITE RIGHT!	562
FIGURE 20-50: EMPLOYEE'S RELATED TRAINING SHOWN IN TRAINING DATAGRIDVIEW	563
FIGURE 20-51: RESULTS OF CLICKING ON A EMPLOYEE WITH A PICTURE - A REMOTINGEXCEPTION IS THROWN	563
FIGURE 20-52: BITMAP CLASS USAGE NOTE	563
FIGURE 20-53: EMPLOYEETRAININGCLIENT APPLICATION WITH EMPLOYEE'S PICTURE DISPLAYED IN THE PICTUREBOX	568
FIGURE 20-54: EMPLOYEE FORM MOCK-UP	569
FIGURE 20-55: TRAINING FORM MOCK-UP	573
FIGURE 20-56: MAIN APPLICATION WINDOW WITH EDIT MENU OPEN TO REVEAL REVISED MENU STRUCTURE	581
FIGURE 20-57: EDIT MENU ITEMS DISABLED	581
FIGURE 20-58: EMPTY EMPLOYEE DATA ENTRY FORM	582
FIGURE 20-59: EMPLOYEE FORM FULLY POPULATE AND SUBMIT BUTTON ENABLED	582
FIGURE 20-60: TRAINING FORM EMPTY AND FILLED	582
FIGURE 21-1: METHOD SIGNATURE FOR OVERLOADED UNARY OPERATOR	591
FIGURE 21-2: METHOD SIGNATURE FOR OVERLOADED UNARY LOGICAL OPERATOR	591
FIGURE 21-3: RESULTS OF RUNNING EXAMPLE 21.2	592
FIGURE 21-4: RESULTS OF RUNNING EXAMPLE 21.4	593
FIGURE 21-5: RESULTS OF RUNNING EXAMPLE 21.6	594
FIGURE 21-6: RESULTS OF RUNNING EXAMPLE 21.9	596
FIGURE 21-7: OVERLOADED BINARY + OPERATOR SIGNATURE THAT OPERATES ON TWO OBJECTS OF TYPE MYTYPE	597
FIGURE 21-8: OVERLOADED BINARY + OPERATOR SIGNATURE THAT OPERATES ON OBJECTS OF MYTYPE AND INTEGER	597
FIGURE 21-9: RESULTS OF RUNNING EXAMPLE 21.11	599
FIGURE 21-10: RESULTS OF RUNNING EXAMPLE 21.13	601
FIGURE 21-11: RESULTS OF RUNNING EXAMPLE 21.15	603
FIGURE 21-12: METHOD SIGNATURE FOR OVERLOADED EQUALITY OPERATOR	604
FIGURE 21-13: COMPILER WARNING – == AND != OPERATORS NEED SPECIAL ATTENTION	606
FIGURE 21-14: RESULTS OF RUNNING EXAMPLE 21.17	606
FIGURE 21-15: METHOD SIGNATURES FOR IMPLICIT AND EXPLICIT CAST OPERATORS	607

FIGURE 21-16: RESULTS OF RUNNING EXAMPLE 21.19	609
FIGURE 21-17: RESULTS OF RUNNING EXAMPLE 21.20	610
FIGURE 22-1: HORIZONTAL AND VERTICAL MEMBER ACCESSIBILITY	617
FIGURE 22-2: RUNNING EXAMPLE 22.2 SEVERAL TIMES	620
FIGURE 22-3: RUNNING MAINAPP IN THE READ MODE	623
FIGURE 22-4: RESULTS OF RUNNING MAINAPP SEVERAL MORE TIMES IN THE APPEND MODE THEN READ MODE	623
FIGURE 22-5: CONCEPT OF A SHALLOW COPY	625
FIGURE 22-6: CONCEPT OF A DEEP COPY	625
FIGURE 22-7: RESULTS OF RUNNING EXAMPLE 22.6	627
FIGURE 22-8: RESULTS OF RUNNING EXAMPLE 22.8	628
FIGURE 22-9: RESULTS OF RUNNING EXAMPLE 22.10	633
FIGURE 22-10: RESULTS OF RUNNING EXAMPLE 22.12	636
FIGURE 22-11: RESULTS OF RUNNING EXAMPLE 22.14	637
FIGURE 23-1: RESULTS OF RUNNING EXAMPLE 23.2	646
FIGURE 23-2: RESULTS OF RUNNING EXAMPLE 23.4	648
FIGURE 23-3: RESULTS OF RUNNING EXAMPLE 23.6	650
FIGURE 23-4: RESULTS OF RUNNING EXAMPLE 24.8	652
FIGURE 23-5: STRONG VS. WEAK TYPES	653
FIGURE 23-6: RESULTS OF RUNNING EXAMPLE 24.12	655
FIGURE 23-7: NAVAL FLEET CLASS INHERITANCE HIERARCHY	657
FIGURE 23-8: RESULTS OF RUNNING EXAMPLE 24.22	661
FIGURE 23-9: TRADITIONAL TOP-DOWN FUNCTIONAL DEPENDENCIES	662
FIGURE 24-1: MEYER'S INHERITANCE TAXONOMY	671
FIGURE 24-2: PERSON-EMPLOYEE INHERITANCE DIAGRAM	673
FIGURE 24-3: REVISED PERSON - EMPLOYEE EXAMPLE	677
FIGURE 24-4: RESULTS OF RUNNING EXAMPLE 24.9	682
FIGURE 25-1: RESULTS OF RUNNING EXAMPLE 25.4	692
FIGURE 25-2: RESULTS OF RUNNING EXAMPLE 25.8	695
FIGURE 25-3: MODEL-VIEW-CONTROLLER PATTERN	695
FIGURE 25-4: RESULTS OF RUNNING EXAMPLE 25.11 AND CLICKING THE "NEXT MESSAGE" BUTTON SEVERAL TIMES	697
FIGURE 25-5: EMPLOYEEMVC PROJECT DIRECTORY STRUCTURE	702
FIGURE 25-6: INTERACTING WITH THE EMPLOYEE MANAGEMENT APPLICATION	721