

# OBJECT-ORIENTED SPECIFICATION AND DESIGN WITH C++

Peter Henderson

*University of Southampton*

---

**McGRAW-HILL BOOK COMPANY**

**London · New York · St Louis · San Francisco · Auckland · Bogotá · Caracas  
Lisbon · Madrid · Mexico · Milan · Montreal · New Delhi · Panama · Paris  
San Juan · São Paulo · Singapore · Sydney · Tokyo · Toronto**

# CONTENTS

---

<b>Preface</b>	<b>ix</b>
<b>Acknowledgements</b>	<b>xiii</b>
<b>1 On what we hope to achieve</b>	<b>1</b>
1.1 Objectives	1
1.2 A view of the software development process	2
1.3 The methods to be described	4
1.4 Outline of the book	6
<b>2 The method applied – a first example</b>	<b>8</b>
2.1 Objectives	8
2.2 Problem statement	8
2.3 Analysis	9
2.4 Design	13
2.5 Implementation	19
2.6 Conclusions	26
<b>3 Object-oriented methods</b>	<b>28</b>
3.1 Introduction	28
3.2 Objects as state holders	29
3.3 Objects as abstract data types	30
3.4 Objects and classes	31
3.5 The augmented entity relationship diagram	32
3.6 Simple inheritance	34
3.7 Collections	37
3.8 Protocol description	38
3.9 The object diagram	43
3.10 Abstract systems	45
3.11 The design stage	46

3.12	Introduction to Enact	47
3.13	Enact is an expression language	48
3.14	Function definition	51
3.15	Assignment statements and statement sequencing	53
3.16	Classes and objects	54
3.17	Simple inheritance	57
3.18	Collections	59
3.19	Lambda expressions	64
3.20	Conclusions	66
3.21	Exercises	66
<b>4</b>	<b>Introducing C++</b>	<b>68</b>
4.1	Introduction	68
4.2	Expressions in C++	69
4.3	Function definitions	72
4.4	Assignment and statements	77
4.5	Objects and classes	80
4.6	Pointers and memory	86
4.7	Inheritance	91
4.8	Collections	97
4.9	The library redone	105
4.10	Conclusions	110
4.11	Exercises	111
<b>5</b>	<b>The method applied – a second example</b>	<b>113</b>
5.1	Introduction	113
5.2	The thermometer problem	115
5.3	User interface objects	125
5.4	A user interface for thermometers	136
5.5	Conclusions	145
5.6	Exercises	146
<b>6</b>	<b>A user interface for the personal library</b>	<b>147</b>
6.1	Introduction	147
6.2	Analysis	149
6.3	Design	153
6.4	Implementation	158
6.5	Conclusions	163
6.6	Exercises	164
<b>7</b>	<b>Persistence</b>	<b>165</b>
7.1	Introduction	165
7.2	Analysis	167
7.3	Design	170
7.4	Implementation	176
7.5	Exercises	185

<b>8 Generic applications</b>	<b>186</b>
8.1 Introduction	186
8.2 A hierarchy editor	187
8.3 A more generic application	197
8.4 On being generic	199
8.5 Exercises	200
<b>9 Some foundational issues</b>	<b>202</b>
9.1 Introduction	202
9.2 Methods of specification	203
9.3 Functional Programming	207
9.4 Object-oriented specification	211
9.5 Advanced C++ concepts	213
9.6 Exercises	216
<b>10 Our methods in context</b>	<b>217</b>
10.1 Introduction	217
10.2 What we set out to achieve	217
10.3 What we actually did, and some alternatives	218
10.4 On what we can conclude, and where next	221
<b>Appendix 1: Enact reference manual</b>	<b>223</b>
A1.1 Getting started	223
A1.2 Functional programming	224
A1.3 Object-oriented programming	226
A1.4 Assignment	227
A1.5 Inheritance	229
A1.6 Operator precedence	230
A1.7 Predefined functions and classes	233
A1.8 Some issues	238
A1.9 Availability	239
<b>Appendix 2: Collection and interface libraries</b>	<b>240</b>
A2.1 Collection classes in Enact	240
A2.2 Source of collect.h	242
A2.3 Source of collect.cpp	244
A2.4 Source of userface.act	247
A2.5 Source of userface.h	249
<b>Appendix 3: Annotated bibliography</b>	<b>252</b>
<b>Index</b>	<b>259</b>