

---

---

# Contents

<b>1</b>	<b>Introduction to the CLOS Model</b>	<b>1</b>
<b>2</b>	<b>Elements of CLOS Programs</b>	<b>5</b>
2.1	Classes and Instances	5
2.2	Slots	6
2.3	Superclasses	7
2.4	Generic Functions	8
2.5	Methods	10
2.6	Method Roles	11
2.7	The Controller of Inheritance	13
2.8	Summary of the CLOS Model	13
2.9	How CLOS Extends COMMON LISP	14
<b>3</b>	<b>Developing a Simple CLOS Program: Locks</b>	<b>17</b>
3.1	Overview of Locking	17
3.2	Defining the Kinds of Objects—Classes	19
3.3	Creating New Objects—Instances	24

3.4	Defining the Interface—Generic Functions	27
3.5	Defining the Implementation—Methods	31
3.6	Specializing the Behavior of Locks	37
3.7	Analyzing the Inheritance of Locks	42
3.8	Extending the Locking Program	45
3.9	How Client Programs Use Locks	53
3.10	Reviewing the Lock Classes	56
3.11	The External and Internal Perspectives	60
3.12	Guidelines on Designing Protocols	62
<b>4</b>	<b>Programming with Methods</b>	<b>65</b>
4.1	Implementation Choices: Methods versus Slots	66
4.2	Programming with Accessors	70
4.3	Multi-Methods	75
4.4	Methods for COMMON LISP Types	82
4.5	Methods for Individual LISP Objects	94
4.6	Summary of Method Inheritance	98
<b>5</b>	<b>Controlling the Generic Dispatch</b>	<b>101</b>
5.1	The Core Framework	101
5.2	Declarative and Imperative Techniques	102
5.3	Around-Methods	102
5.4	Calling a Shadowed Primary Method	105
5.5	Using a Different Method Combination Type	105
5.6	Built-in Method Combination Types	107
5.7	Defining a New Method Combination Type	109
5.8	Guidelines on Controlling the Generic Dispatch	111
5.9	Summary of the Generic Dispatch Procedure	112
5.10	Summary of the Standard Method Combination Type	113
<b>6</b>	<b>Class Inheritance</b>	<b>117</b>
6.1	Inheritance from Default Classes	117
6.2	The Class Precedence List	118
6.3	Guidelines on Designing Class Organizations	125
6.4	Inheritance of Slots and Slot Options	127
6.5	Guidelines on Using Inheritance of Slot Options	129
<b>7</b>	<b>Defining CLOS Elements</b>	<b>131</b>
7.1	Order of Defining CLOS Elements	131
7.2	Congruent Lambda-Lists	132

7.3	Lisp Objects Representing CLOS Elements	133
7.4	Mapping Between Names and Objects	134
7.5	Removing Generic Functions and Methods	136
<b>8</b>	<b>Redefining CLOS Elements</b>	<b>139</b>
8.1	Redefining Classes	140
8.2	Redefining Methods and Generic Functions	143
8.3	Example of Redefining CLOS Elements	144
8.4	Changing the Class of an Instance	151
<b>9</b>	<b>Creating and Initializing Instances</b>	<b>155</b>
9.1	Arguments to make-instance	155
9.2	Summary of What make-instance Does	156
9.3	Controlling Initialization with defclass Options	157
9.4	Controlling Initialization with Methods	159
9.5	Initialization Arguments	160
9.6	Constructors	163
<b>10</b>	<b>A Procedural Definition: Initialization</b>	<b>165</b>
10.1	Examples of Procedural Definitions	165
10.2	Isolating Work Shared Among Procedures	167
10.3	Filling Unbound Slots with Initforms	168
10.4	Performing Initialization by Initargs	168
10.5	Specializing a Portion of the Procedure	169
10.6	Declaring Initarg Names as Valid	170
<b>11</b>	<b>Developing an Advanced CLOS Program: Streams</b>	<b>171</b>
11.1	Overview of Streams	172
11.2	Design of Our Stream Foundation	175
11.3	Experimenting with the Stream Example	179
11.4	Directional Streams	180
11.5	Tape Streams	185
11.6	Disk Streams	190
11.7	Character Streams	199
11.8	Byte Streams	202
11.9	Instantiable Streams	205
11.10	A Procedural Definition for Creating Streams	207
11.11	Summary of Techniques Used in Streams	212
11.12	Documenting the Implementation of Streams	212

<b>12 Highlights of CLOS</b>	<b>215</b>
12.1 Designing a Standard Object-Oriented Paradigm	216
12.2 Summary of the Programmer Interface	216
12.3 Exploring Alternative Paradigms	218
12.4 A Nongoal: Automatic Protocol Support	218
12.5 The CLOS Metaobject Protocol	219
<b>Appendix A Glossary of CLOS Terminology</b>	<b>223</b>
<b>Appendix B Syntax of CLOS Operators</b>	<b>233</b>
<b>Appendix C CLOS Operators Not Documented in This Book</b>	<b>253</b>
<b>Bibliography</b>	<b>257</b>
<b>Index</b>	<b>259</b>