

# Contents

<b>Preface</b>	<b>vii</b>
<b>1 OBJECT-ORIENTED CONCEPTS</b>	<b>1</b>
1.1 Application Domains and Solution Domains . . . . .	2
1.2 Profile of Present Software Systems . . . . .	4
1.3 Towards an Object-Oriented Approach . . . . .	6
1.3.1 Abstract Data Types . . . . .	8
1.3.2 Evolution of Abstraction in Programming Languages . . . . .	11
1.3.3 Comparison of Selected Object-Oriented Languages . . . . .	14
1.4 Characterization of an Object-Oriented Model . . . . .	22
1.4.1 Objects . . . . .	24
1.4.2 Classes . . . . .	24
1.4.3 Inheritance . . . . .	26
1.5 Philosophy of Object-Oriented Design . . . . .	29
1.5.1 Domain Analysis . . . . .	32
1.5.2 Reuse of Software . . . . .	34
1.5.3 Software Life Cycle Models . . . . .	38
1.6 Summary . . . . .	46
<b>2 OBJECT-ORIENTED METHODOLOGIES</b>	<b>49</b>
2.1 Classification of Methodologies . . . . .	50
2.1.1 Classification of Existing Object-Oriented Methodologies . . . . .	55
2.2 Survey of Existing Methodologies . . . . .	57

2.2.1	Booch Methodologies and Influences . . . . .	58
2.2.2	Responsibility-Driven Design . . . . .	70
2.2.3	OOA . . . . .	73
2.2.4	OMT . . . . .	75
2.2.5	OOSA . . . . .	77
2.2.6	OSA . . . . .	78
2.2.7	Objectory . . . . .	78
2.2.8	Pun and Winder Methodology . . . . .	80
2.2.9	JSD . . . . .	81
2.2.10	Graphical Notations for Implementation . . .	82
2.3	Results of the Classification . . . . .	83
<b>3</b>	<b>OBJECT-ORIENTED DESIGN</b>	<b>87</b>
3.1	The Context of MOOD . . . . .	89
3.2	Steps for Object-Oriented Design . . . . .	92
3.2.1	Representation of the Design Model . . . . .	93
3.2.2	Identification of Components . . . . .	95
3.2.3	Identification of Classes . . . . .	96
3.2.4	Identification of Inheritance . . . . .	98
3.2.5	Identification of Objects . . . . .	104
3.2.6	Identification of Software Behaviour . . . . .	108
3.3	Suitability of MOOD . . . . .	112
3.3.1	Partitioning Issues . . . . .	112
3.3.2	Inheritance within MOOD . . . . .	113
3.3.3	Representation of Control Flow . . . . .	114
3.3.4	Object Interactions . . . . .	115
3.4	A CASE Environment for MOOD . . . . .	116
3.4.1	MOOD Tools . . . . .	119
3.5	Observations on MOOD . . . . .	122
<b>4</b>	<b>REUSABILITY AND LIFE CYCLE ISSUES</b>	<b>127</b>
4.1	Reusability During Object-Oriented Design . . . . .	128
4.1.1	Relationships Between Components . . . . .	129
4.1.2	Reusability Process . . . . .	132
4.1.3	Lifetime of Reusable Components . . . . .	133

4.2	Object-Oriented Software Life Cycle Model . . . . .	137
4.2.1	Role of the Knowledge about the Application Domain . . . . .	137
4.2.2	A Seamless Software Life Cycle . . . . .	139
4.3	Software Life Cycle Issues . . . . .	146
4.3.1	Types of Reusable Components . . . . .	146
4.3.2	Software Process Issues . . . . .	147
4.3.3	Mechanisms Prevalent in Each Life Cycle Phase	147
4.3.4	Percentage of Time per Each Development Stage	148
4.4	Comments on the Software Process . . . . .	149
<b>5</b>	<b>SOFTWARE MAINTENANCE CONCEPTS</b>	<b>153</b>
5.1	Software Maintenance Process . . . . .	154
5.2	Supporting Maintenance Technology . . . . .	157
5.2.1	Reverse Engineering Environments . . . . .	159
5.3	Documentation for Software Maintenance . . . . .	163
5.3.1	Redocumentation . . . . .	166
5.3.2	Support Available for Software Documentation	168
5.3.3	Documentation Tools for Software Maintenance	169
5.4	Software Configuration Management . . . . .	170
5.4.1	Functions of SCM . . . . .	171
5.4.2	Automation of SCM . . . . .	173
5.4.3	Application of SCM to Software Maintenance	176
5.5	Summary . . . . .	177
<b>6</b>	<b>SOFTWARE MAINTENANCE MODELS</b>	<b>181</b>
6.1	Managing the Software Life Cycle with Maintenance .	181
6.2	Existing Software Maintenance Models . . . . .	183
6.2.1	Early Software Maintenance Models . . . . .	184
6.2.2	Middle Software Maintenance Models . . . . .	186
6.2.3	Recent Software Maintenance Models . . . . .	190
6.3	Software Process Modelling for Maintenance . . . . .	194
6.4	Characteristics of Software Maintenance Modelling . .	197
6.5	Summary . . . . .	199

<b>7</b>	<b>MAINTENANCE OF LEGACY SYSTEMS</b>	<b>201</b>
7.1	Objectives of COMFORM . . . . .	201
7.2	Software Maintenance Model . . . . .	204
7.2.1	Change Request . . . . .	207
7.2.2	Change Evaluation . . . . .	208
7.2.3	Maintenance Design Specification . . . . .	212
7.2.4	Maintenance Design Redocumentation . . . . .	215
7.2.5	Maintenance Implementation . . . . .	216
7.2.6	System Release . . . . .	217
7.3	SCM Discipline Applied to COMFORM . . . . .	219
7.3.1	Software Configuration Identification . . . . .	219
7.3.2	Software Configuration Control . . . . .	220
7.3.3	Software Configuration Status Accounting . . . . .	220
7.3.4	Software Configuration Auditing . . . . .	223
7.4	Version Control in COMFORM . . . . .	226
7.5	Remarks on COMFORM . . . . .	230
<b>8</b>	<b>OBJECT-ORIENTED SOFTWARE EVOLUTION</b>	<b>233</b>
8.1	Object-Orientation for Software Maintenance . . . . .	233
8.1.1	Definition of SMM Forms Using Class Hierarchies	235
8.1.2	Inheritance Mechanism for Version Control of Forms . . . . .	238
8.2	On MOOD and COMFORM . . . . .	240
8.3	The Future . . . . .	242
8.4	In Conclusion . . . . .	245
	<b>References</b>	<b>249</b>
	<b>Index</b>	<b>259</b>