

# Contents

<b>Chapter 1</b>	<b>Preliminaries</b>	<b>25</b>
1.1	Reasons for Studying Concepts of Programming Languages.....	26
1.2	Programming Domains.....	29
1.3	Language Evaluation Criteria.....	30
1.4	Influences on Language Design.....	41
1.5	Language Categories.....	44
1.6	Language Design Trade-Offs.....	45
1.7	Implementation Methods.....	46
1.8	Programming Environments .....	53
	Summary • Review Questions • Problem Set.....	54
<b>Chapter 2</b>	<b>Evolution of the Major Programming Languages</b>	<b>57</b>
2.1	Zuse's Plankalkül .....	60
2.2	Pseudocodes.....	61
2.3	The IBM 704 and Fortran .....	64
2.4	Functional Programming: Lisp.....	69
2.5	The First Step Toward Sophistication: ALGOL 60.....	74
2.6	Computerizing Business Records: COBOL.....	80
2.7	The Beginnings of Timesharing: Basic.....	85
	Interview: <b>ALAN COOPER—User Design and Language Design</b> .....	88
2.8	Everything for Everybody: PL/I.....	90
2.9	Two Early Dynamic Languages: APL and SNOBOL.....	93
2.10	The Beginnings of Data Abstraction: SIMULA 67.....	94
2.11	Orthogonal Design: ALGOL 68.....	95
2.12	Some Early Descendants of the ALGOLs.....	97

2.13	Programming Based on Logic: Prolog .....	101
2.14	History's Largest Design Effort: Ada .....	103
2.15	Object-Oriented Programming: Smalltalk .....	107
2.16	Combining Imperative and Object-Oriented Features: C++ .....	109
2.17	An Imperative-Based Object-Oriented Language: Java .....	112
2.18	Scripting Languages .....	115
2.19	The Flagship .NET Language: C# .....	122
2.20	Markup-Programming Hybrid Languages .....	124
	Summary • Bibliographic Notes • Review Questions • Problem Set • Programming Exercises .....	126
<b>Chapter 3</b>	<b>Describing Syntax and Semantics</b>	<b>133</b>
3.1	Introduction .....	134
3.2	The General Problem of Describing Syntax .....	135
3.3	Formal Methods of Describing Syntax .....	137
3.4	Attribute Grammars .....	152
	<i>History Note</i> .....	152
3.5	Describing the Meanings of Programs: Dynamic Semantics .....	158
	<i>History Note</i> .....	166
	Summary • Bibliographic Notes • Review Questions • Problem Set .....	179
<b>Chapter 4</b>	<b>Lexical and Syntax Analysis</b>	<b>185</b>
4.1	Introduction .....	186
4.2	Lexical Analysis .....	187
4.3	The Parsing Problem .....	195
4.4	Recursive-Descent Parsing .....	199
4.5	Bottom-Up Parsing .....	207
	Summary • Review Questions • Problem Set • Programming Exercises .....	215
<b>Chapter 5</b>	<b>Names, Bindings, and Scopes</b>	<b>221</b>
5.1	Introduction .....	222
5.2	Names .....	223

	<b>History Note</b> .....	223
<b>5.3</b>	Variables.....	224
<b>5.4</b>	The Concept of Binding .....	227
<b>5.5</b>	Scope.....	235
<b>5.6</b>	Scope and Lifetime.....	246
<b>5.7</b>	Referencing Environments .....	247
<b>5.8</b>	Named Constants.....	248
	Summary • Review Questions • Problem Set • Programming Exercises .....	251

**Chapter 6 Data Types 259**

<b>6.1</b>	Introduction.....	260
<b>6.2</b>	Primitive Data Types.....	262
<b>6.3</b>	Character String Types.....	266
	<b>History Note</b> .....	267
<b>6.4</b>	Enumeration Types .....	271
<b>6.5</b>	Array Types.....	274
	<b>History Note</b> .....	275
	<b>History Note</b> .....	275
<b>6.6</b>	Associative Arrays .....	285
	Interview: <b>ROBERTO IERUSALIMSKY—Lua</b> .....	286
<b>6.7</b>	Record Types .....	289
<b>6.8</b>	Tuple Types.....	292
<b>6.9</b>	List Types.....	294
<b>6.10</b>	Union Types .....	296
<b>6.11</b>	Pointer and Reference Types .....	299
	<b>History Note</b> .....	302
<b>6.12</b>	Type Checking.....	311
<b>6.13</b>	Strong Typing.....	312
<b>6.14</b>	Type Equivalence.....	313
<b>6.15</b>	Theory and Data Types.....	317
	Summary • Bibliographic Notes • Review Questions • Problem Set • Programming Exercises .....	319

<b>Chapter 7</b>	<b>Expressions and Assignment Statements</b>	<b>325</b>
7.1	Introduction.....	326
7.2	Arithmetic Expressions.....	326
7.3	Overloaded Operators.....	335
7.4	Type Conversions.....	337
	<b>History Note</b> .....	339
7.5	Relational and Boolean Expressions.....	340
	<b>History Note</b> .....	340
7.6	Short-Circuit Evaluation.....	342
7.7	Assignment Statements.....	343
	<b>History Note</b> .....	347
7.8	Mixed-Mode Assignment.....	348
	Summary • Review Questions • Problem Set • Programming Exercises....	348
<b>Chapter 8</b>	<b>Statement-Level Control Structures</b>	<b>353</b>
8.1	Introduction.....	354
8.2	Selection Statements.....	356
8.3	Iterative Statements.....	367
8.4	Unconditional Branching.....	379
	<b>History Note</b> .....	379
8.5	Guarded Commands.....	380
8.6	Conclusions.....	382
	Summary • Review Questions • Problem Set • Programming Exercises....	383
<b>Chapter 9</b>	<b>Subprograms</b>	<b>389</b>
9.1	Introduction.....	390
9.2	Fundamentals of Subprograms.....	390
9.3	Design Issues for Subprograms.....	398
9.4	Local Referencing Environments.....	399
9.5	Parameter-Passing Methods.....	401
	<b>History Note</b> .....	409

	<b>History Note</b> .....	409
<b>9.6</b>	Parameters That Are Subprograms.....	417
	<b>History Note</b> .....	419
<b>9.7</b>	Calling Subprograms Indirectly.....	419
<b>9.8</b>	Design Issues for Functions .....	421
<b>9.9</b>	Overloaded Subprograms.....	423
<b>9.10</b>	Generic Subprograms.....	424
<b>9.11</b>	User-Defined Overloaded Operators .....	430
<b>9.12</b>	Closures .....	430
<b>9.13</b>	Coroutines .....	432
	Summary • Review Questions • Problem Set • Programming Exercises.....	435

## **Chapter 10 Implementing Subprograms 441**

<b>10.1</b>	The General Semantics of Calls and Returns.....	442
<b>10.2</b>	Implementing “Simple” Subprograms.....	443
<b>10.3</b>	Implementing Subprograms with Stack-Dynamic Local Variables.....	445
<b>10.4</b>	Nested Subprograms .....	453
<b>10.5</b>	Blocks.....	460
<b>10.6</b>	Implementing Dynamic Scoping .....	461
	Summary • Review Questions • Problem Set • Programming Exercises.....	465

## **Chapter 11 Abstract Data Types and Encapsulation Constructs 471**

<b>11.1</b>	The Concept of Abstraction.....	472
<b>11.2</b>	Introduction to Data Abstraction.....	473
<b>11.3</b>	Design Issues for Abstract Data Types.....	476
<b>11.4</b>	Language Examples.....	477
	Interview: <b>BJARNE STROUSTRUP—C++: Its Birth, Its Ubiquitousness, and Common Criticisms</b> .....	478
<b>11.5</b>	Parameterized Abstract Data Types .....	496
<b>11.6</b>	Encapsulation Constructs.....	500
<b>11.7</b>	Naming Encapsulations .....	504
	Summary • Review Questions • Problem Set • Programming Exercises.....	507

<b>Chapter 12</b>	<b>Support for Object-Oriented Programming</b>	<b>513</b>
12.1	Introduction.....	514
12.2	Object-Oriented Programming .....	515
12.3	Design Issues for Object-Oriented Languages.....	519
12.4	Support for Object-Oriented Programming in Specific Languages .....	524
	Interview: <b>BJARNE STROUSTRUP—On Paradigms and Better Programming</b> .....	528
12.5	Implementation of Object-Oriented Constructs.....	552
12.6	Reflection.....	555
	Summary • Review Questions • Problem Set • Programming Exercises.....	561
<b>Chapter 13</b>	<b>Concurrency</b>	<b>567</b>
13.1	Introduction.....	568
13.2	Introduction to Subprogram-Level Concurrency.....	573
13.3	Semaphores.....	578
13.4	Monitors .....	583
13.5	Message Passing .....	585
13.6	Ada Support for Concurrency.....	586
13.7	Java Threads .....	594
13.8	C# Threads .....	604
13.9	Concurrency in Functional Languages.....	609
13.10	Statement-Level Concurrency.....	612
	Summary • Bibliographic Notes • Review Questions • Problem Set • Programming Exercises.....	614
<b>Chapter 14</b>	<b>Exception Handling and Event Handling</b>	<b>621</b>
14.1	Introduction to Exception Handling .....	622
	History Note .....	626
14.2	Exception Handling in C++.....	628
14.3	Exception Handling in Java.....	632
14.4	Exception Handling in Python and Ruby.....	639
14.5	Introduction to Event Handling .....	642
14.6	Event Handling with Java .....	643

14.7	Event Handling in C# .....	647
	Summary • Bibliographic Notes • Review Questions • Problem Set • Programming Exercises .....	650
<b>Chapter 15</b>	<b>Functional Programming Languages</b>	<b>657</b>
15.1	Introduction .....	658
15.2	Mathematical Functions .....	659
15.3	Fundamentals of Functional Programming Languages .....	662
15.4	The First Functional Programming Language: Lisp .....	663
15.5	An Introduction to Scheme .....	667
15.6	Common Lisp .....	685
15.7	ML .....	687
15.8	Haskell .....	692
15.9	F# .....	697
15.10	Support for Functional Programming in Primarily Imperative Languages .....	700
15.11	A Comparison of Functional and Imperative Languages .....	703
	Summary • Bibliographic Notes • Review Questions • Problem Set • Programming Exercises .....	705
<b>Chapter 16</b>	<b>Logic Programming Languages</b>	<b>713</b>
16.1	Introduction .....	714
16.2	A Brief Introduction to Predicate Calculus .....	714
16.3	Predicate Calculus and Proving Theorems .....	718
16.4	An Overview of Logic Programming .....	720
16.5	The Origins of Prolog .....	722
16.6	The Basic Elements of Prolog .....	722
16.7	Deficiencies of Prolog .....	737
16.8	Applications of Logic Programming .....	743
	Summary • Bibliographic Notes • Review Questions • Problem Set • Programming Exercises .....	744
	<b>Bibliography .....</b>	<b>749</b>
	<b>Index .....</b>	<b>761</b>