

# Contents

<b>Chapter 1</b>	<b>Preliminaries</b>	<b>19</b>
1.1	Reasons for Studying Concepts of Programming Languages .....	20
1.2	Programming Domains .....	23
1.3	Language Evaluation Criteria .....	25
1.4	Influences on Language Design .....	37
1.5	Language Categories .....	40
1.6	Language Design Trade-Offs .....	42
1.7	Implementation Methods .....	43
1.8	Programming Environments .....	50
	Summary • Review Questions • Problem Set .....	51
<b>Chapter 2</b>	<b>Evolution of the Major Programming Languages</b>	<b>55</b>
2.1	Zuse's Plankalkül .....	56
2.2	Minimal Hardware Programming: PseudoCodes .....	59
2.3	The IBM 704 and Fortran .....	62
2.4	Functional Programming: LISP .....	67
2.5	The First Step Toward Sophistication: ALGOL 60 .....	72
2.6	Computerizing Business Records: COBOL .....	78
2.7	The Beginnings of Timesharing: BASIC .....	83
	Interview: <b>ALAN COOPER—User Design and Language Design</b> .....	86
2.8	Everything for Everybody: PL/I .....	88
2.9	Two Early Dynamic Languages: APL and SNOBOL .....	91
2.10	The Beginnings of Data Abstraction: SIMULA 67 .....	92
2.11	Orthogonal Design: ALGOL 68 .....	93

2.12	Some Early Descendants of the ALGOLs .....	95
2.13	Programming Based on Logic: Prolog .....	99
2.14	History's Largest Design Effort: Ada .....	101
2.15	Object-Oriented Programming: Smalltalk .....	105
2.16	Combining Imperative and Object-Oriented Features: C++ .....	108
2.17	An Imperative-Based Object-Oriented Language: Java .....	111
2.18	Scripting Languages .....	115
2.19	A C-Based Language for the New Millennium: C# .....	121
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
	History Note .....	153
3.5	Describing the Meanings of Programs: Dynamic Semantics .....	159
	History Note .....	174
	Summary • Bibliographic Notes • Review Questions • Problem Set .....	181
<b>Chapter 4</b>	<b>Lexical and Syntax Analysis</b>	<b>189</b>
4.1	Introduction .....	190
4.2	Lexical Analysis .....	191
4.3	The Parsing Problem .....	200
4.4	Recursive-Descent Parsing .....	203
4.5	Bottom-Up Parsing .....	212
	Summary • Review Questions • Problem Set • Programming Exercises.....	220
<b>Chapter 5</b>	<b>Names, Bindings, and Scopes</b>	<b>225</b>
5.1	Introduction .....	226
5.2	Names .....	226

	History Note .....	227
	History Note .....	228
<b>5.3</b>	<b>Variables .....</b>	<b>229</b>
	History Note .....	230
<b>5.4</b>	<b>The Concept of Binding .....</b>	<b>232</b>
	<b>Interview: RASMUS LERDORF—Scripting Languages and Other Examples of Slick Solutions .....</b>	<b>236</b>
<b>5.5</b>	<b>Scope .....</b>	<b>243</b>
	History Note .....	245
<b>5.6</b>	<b>Scope and Lifetime .....</b>	<b>252</b>
<b>5.7</b>	<b>Referencing Environments .....</b>	<b>253</b>
<b>5.8</b>	<b>Named Constants .....</b>	<b>255</b>
	<b>Summary • Review Questions • Problem Set • Programming Exercises ...</b>	<b>257</b>
<b>Chapter 6</b>	<b>Data Types .....</b>	<b>265</b>
<b>6.1</b>	<b>Introduction .....</b>	<b>266</b>
<b>6.2</b>	<b>Primitive Data Types .....</b>	<b>268</b>
<b>6.3</b>	<b>Character String Types .....</b>	<b>272</b>
	History Note .....	273
<b>6.4</b>	<b>User-Defined Ordinal Types .....</b>	<b>277</b>
<b>6.5</b>	<b>Array Types .....</b>	<b>281</b>
	History Note .....	282
	History Note .....	285
<b>6.6</b>	<b>Associative Arrays .....</b>	<b>295</b>
	<b>Interview: ROBERTO IERUSALIMSCHY—Lua .....</b>	<b>298</b>
<b>6.7</b>	<b>Record Types .....</b>	<b>300</b>
<b>6.8</b>	<b>Union Types .....</b>	<b>305</b>
<b>6.9</b>	<b>Pointer and Reference Types .....</b>	<b>309</b>
	History Note .....	312
<b>6.10</b>	<b>Type Checking .....</b>	<b>322</b>
<b>6.11</b>	<b>Strong Typing .....</b>	<b>323</b>
<b>6.12</b>	<b>Type Equivalence .....</b>	<b>324</b>

6.13	Theory and Data Types .....	328
	Summary • Bibliographic Notes • Review Questions • Problem Set • Programming Exercises .....	330
<b>Chapter 7</b>	<b>Expressions and Assignment Statements</b>	<b>337</b>
7.1	Introduction .....	338
7.2	Arithmetic Expressions .....	339
7.3	Overloaded Operators .....	348
7.4	Type Conversions .....	349
	History Note .....	352
7.5	Relational and Boolean Expressions .....	353
	History Note .....	353
7.6	Short-Circuit Evaluation .....	355
7.7	Assignment Statements .....	357
	History Note .....	361
7.8	Mixed-Mode Assignment .....	361
	Summary • Review Questions • Problem Set • Programming Exercises ....	362
<b>Chapter 8</b>	<b>Statement-Level Control Structures</b>	<b>367</b>
8.1	Introduction .....	368
8.2	Selection Statements .....	370
	History Note .....	370
	History Note .....	372
8.3	Iterative Statements .....	383
	History Note .....	383
	Interview: LARRY WALL—Part 1: Linguistics and the Birth of Perl .....	388
	History Note .....	396
8.4	Unconditional Branching .....	398
8.5	Guarded Commands .....	399
8.6	Conclusions .....	401
	Summary • Review Questions • Problem Set • Programming Exercises.....	402

<b>Chapter 9</b>	<b>Subprograms</b>	<b>409</b>
9.1	Introduction .....	410
9.2	Fundamentals of Subprograms .....	410
9.3	Design Issues for Subprograms .....	421
9.4	Local Referencing Environments .....	422
9.5	Parameter-Passing Methods .....	424
	<b>Interview: LARRY WALL—Part 2: Scripting Languages in General and Perl in Particular</b> .....	<b>426</b>
	History Note .....	434
	History Note .....	434
	History Note .....	438
9.6	Parameters That Are Subprograms .....	445
	History Note .....	447
9.7	Overloaded Subprograms .....	447
9.8	Generic Subprograms .....	449
9.9	Design Issues for Functions .....	456
9.10	User-Defined Overloaded Operators .....	457
9.11	Coroutines .....	458
	History Note .....	458
	Summary • Review Questions • Problem Set • Programming Exercises.....	460
<b>Chapter 10</b>	<b>Implementing Subprograms</b>	<b>467</b>
10.1	The General Semantics of Calls and Returns .....	468
10.2	Implementing “Simple” Subprograms .....	459
10.3	Implementing Subprograms with Stack-Dynamic Local Variables .....	471
10.4	Nested Subprograms .....	480
	<b>Interview: NIKLAUS WIRTH—Keeping It Simple</b> .....	<b>482</b>
10.5	Blocks .....	488
10.6	Implementing Dynamic Scoping .....	490
	Summary • Review Questions • Problem Set • Programming Exercises.....	493

<b>Chapter 11</b>	<b>Abstract Data Types and Encapsulation Constructs</b>	<b>499</b>
11.1	The Concept of Abstraction .....	500
11.2	Introduction to Data Abstraction .....	501
11.3	Design Issues for Abstract Data Types .....	504
11.4	Language Examples .....	505
	<b>Interview: BJARNE STROUSTRUP—C++: Its Birth, Its Ubiquitousness, and Common Criticisms</b> .....	508
11.5	Parameterized Abstract Data Types .....	522
11.6	Encapsulation Constructs .....	526
11.7	Naming Encapsulations .....	530
	Summary • Review Questions • Problem Set • Programming Exercises ....	535
<b>Chapter 12</b>	<b>Support for Object-Oriented Programming</b>	<b>541</b>
12.1	Introduction .....	542
12.2	Object-Oriented Programming .....	542
12.3	Design Issues for Object-Oriented Languages .....	545
12.4	Support for Object-Oriented Programming in Smalltalk .....	550
12.5	Support for Object-Oriented Programming in C++ .....	553
	<b>Interview: BJARNE STROUSTRUP—On Paradigms and Better Programming</b> .....	554
12.6	Support for Object-Oriented Programming in Java .....	564
12.7	Support for Object-Oriented Programming in C# .....	568
12.8	Support for Object-Oriented Programming in Ada 95 .....	570
12.9	Support for Object-Oriented Programming in Ruby.....	575
12.10	Implementation of Object-Oriented Constructs .....	578
	Summary • Review Questions • Problem Set • Programming Exercises.....	582
<b>Chapter 13</b>	<b>Concurrency</b>	<b>587</b>
13.1	Introduction .....	588
13.2	Introduction to Subprogram-Level Concurrency .....	592
	History Note .....	595
13.3	Semaphores .....	596

	13.4	Monitors .....	601
	13.5	Message Passing .....	603
	13.6	Ada Support for Concurrency .....	605
	13.7	Java Threads .....	616
	13.8	C# Threads .....	624
	13.9	Statement-Level Concurrency .....	626
		Summary • Bibliographic Notes • Review Questions • Problem Set • Programming Exercises .....	628
<b>Chapter 14</b>		<b>Exception Handling and Event Handling</b>	<b>633</b>
	14.1	Introduction to Exception Handling .....	634
		History Note .....	638
	14.2	Exception Handling in Ada .....	640
	14.3	Exception Handling in C++ .....	647
	14.4	Exception Handling in Java .....	652
		Interview: <b>JAMES GOSLING—The Birth of Java</b> .....	654
	14.5	Introduction to Event Handling .....	661
	14.6	Event Handling with Java .....	662
		Summary • Bibliographic Notes • Review Questions • Problem Set • Programming Exercises .....	668
<b>Chapter 15</b>		<b>Functional Programming Languages</b>	<b>673</b>
	15.1	Introduction .....	674
	15.2	Mathematical Functions .....	675
	15.3	Fundamentals of Functional Programming Languages .....	677
	15.4	The First Functional Programming Language: LISP .....	679
	15.5	An Introduction to Scheme .....	682
	15.6	COMMON LISP .....	700
	15.7	ML .....	700
	15.8	Haskell .....	704
	15.9	Applications of Functional Languages .....	709
	15.10	A Comparison of Functional and Imperative Languages .....	710
		Summary • Bibliographic Notes • Review Questions • Problem Set • Programming Exercises .....	712

<b>Chapter 16</b>	<b>Logic Programming Languages</b>	<b>719</b>
16.1	Introduction .....	720
16.2	A Brief Introduction to Predicate Calculus .....	720
16.3	Predicate Calculus and Proving Theorems .....	724
16.4	An Overview of Logic Programming .....	726
16.5	The Origins of Prolog .....	728
16.6	The Basic Elements of Prolog .....	729
16.7	Deficiencies of Prolog .....	743
16.8	Applications of Logic Programming .....	749
	Summary • Bibliographic Notes • Review Questions • Problem Set • Programming Exercises.....	751
	<b>Bibliography</b> .....	<b>755</b>
	<b>Index</b> .....	<b>765</b>