

Preface

The **Java Jumpstart** pocketbook was inspired from the experiences of the first author in teaching, training and mentoring the programming language Java to professional developers over several years. From these experiences, two principle objectives became clear. First, skilled programmers, especially those in C and C++, wanted a concise, straightforward introduction to Java that enabled them to begin programming quickly and effectively. Wading through hundreds of pages of text was simply not a feasible option. This small text aims to satisfy that audience. Second, Java is grounded in object technology. This technology is a powerful alternative to the traditional methods of procedural programming and is built on two guiding principles: 1) to clarify responsibility by encapsulating data and function in objects and 2) to encourage reuse through composition and inheritance. This text also aims to provide a basic understanding of object-oriented concepts and how they are integrated into the Java programming language.

Why Read This Book

Those new to Java will appreciate a succinct, hands-on approach to teaching the language, and the sound explanations of how object-oriented concepts are integrated into Java. Experienced programmers will improve their mastery of object-oriented concepts and the notation of the Unified Modeling Language (UML).

Who This Book is For

The authors assume only that the reader has a good knowledge of C and basic data structures (it helps to have a basic knowledge of C++). The text leverages this background to teach the core and advanced concepts of object-oriented programming in Java.

Organization of This Book

The first two chapters are a general introduction to various concepts of the Java programming language, the object-oriented approach and the Unified Modeling Language (UML). Subsequent chapters systematically cover the essential features of the language.

Java Language Fundamentals presents the specific requirements, the main features, the basic elements, and the program structure of the Java programming language.

Object-Oriented Concepts in UML and Java reviews the fundamental concepts of object-oriented (OO) technology with their corresponding Java and UML notations.

Expressions, Assignments, and Operators presents the expressions, assignments and all the various operations that can be performed on primitive types and objects.

Statements, Flow Control and Exceptions covers the basic statements and all the control flow statements. It also describes the use of the exception mechanism for robust error management.

Classes and Access Control presents in detail the fundamental unit of programming in Java: the class. The access control to classes as well as the visibility of their members (fields and methods) is also explained.

Abstract Classes and Interfaces presents how to reuse a class by inheritance. It also introduces abstract classes and interfaces as a way to capture your operations (behavior) and to enable their implementations.

About the Authors

Michel de Champlain is the President and Principal Architect of DeepObjectKnowledge Inc., a firm that provides industry with mentoring and training support in object technologies. Michel holds a Ph.D. in Software Engineering from the Ecole Polytechnique de Montreal, and has held university appointments at the Royal Military College in St. Jean, the University of Canterbury in New Zealand, and Concordia University in Montreal. Working in close collaboration with industry as well as academia, Michel has trained thousands of people throughout Canada, the United States, Europe, and down under in object-oriented analysis, design and implementation. His current research interests include object-oriented languages, frameworks, design patterns, compilers, virtual machines, and real-time microkernels.

—*Michel de Champlain, Ph.D.*
DeepObjectKnowledge, Inc.
mdec@DeepObjectKnowledge.com

Brian G. Patrick is an Associate Professor in Computer Science/Studies at Trent University in Peterborough, Ontario. He first met Michel as a colleague at the Royal Military College in St. Jean, and has developed a close working relationship with Michel over the years. Brian earned his Ph.D. in Computer Science from McGill University in Montreal where he later completed an M.B.A. in Finance and International Business. His research interests have included heuristic search, parallel algorithms, and software reuse. He is currently investigating job scheduling schemes for parallel applications.

—*Brian G. Patrick, Ph.D.*
Trent University
bpatrick@trentu.ca

Table of Contents

Java Language Fundamentals	1
What Is Java?	2
Conventions for Syntax	3
Basic Elements	4
Comments	5
Identifiers	6
Keywords	6
Literals	7
Operators and Punctuators	8
Traditional Language Features	8
Named Constants	9
Data Types	9
Understanding the Program Structure	10
How to Compile and Run Programs	13
Running the Compiler: javac	13
Running the Interpreter: java	13
Accessing Command-Line Arguments	13
Object-Oriented Concepts in UML and Java	15
What Is an Abstraction?	16
Defining Classes	16
Protecting the Data: Encapsulation	17
Defining Class Members: Data and Behavior	17
Creating and Handling Objects	19
Creating Objects: Instantiation	19
Accessing and Invoking Members	22
Messages	22

Invoking Instance Methods	23
Invoking Class Methods	24
Accessing Instance Variables	25
Accessing Class Variables	25
The String Class	25
Invoking String Methods	25
concat and substring Methods	26
Overloading and Cascading Methods	27
Aggregation and Inheritance	28
Reusing Classes: Inheritance	28
Reusing Classes: Aggregation	29
Aggregation vs Inheritance	29
Packages	30
Packages and Directory Path	33
Application Programming Interface	34
Importing Classes	35
Using Streams for I/O	36
Expressions, Assignments, and Operators	39
Operator Precedence and Associativity	40
Explicit and Implicit Casts	41
Widening Casts	42
Narrowing Casts	42
Assignment Operator	43
Simple Assignment	43
Multiple Assignments	44
Conditional Operators	45
Logical Operators	46
Bitwise Operators	47
Binary Bitwise Operators	47
Bitwise Operators as Boolean Logical Operators	49
Bitwise Extended Assignment Operators	49
Equality Operators	51
Primitive Data Type Equality	51
Object Reference Equality	52
Object Value Equality	52
Relational Operators	53
Shift Operators	55
Arithmetic Operators	56

Multiplicative Operators	56
Additive Operators	57
Unary Operators	58
Arithmetic Extended Assignment Operators	59
Prefix and Postfix Operators	59
Boolean Expressions	60
Access Array, new and instanceof Operators	61
Statements, Flow Control and Exceptions	63
Statements	64
Expression Statements	64
Declaration Statements	64
Blocks	65
Flow Control Statements	65
Selection Statements	65
Iteration Statements	68
Transfer Statements	70
Using Exceptions for Error Handling	72
What is an Exception?	73
Using Exception Handling	74
The throws Clause	74
The throw Statement	75
The try-catch Statement	76
Handling Multiple Exceptions	80
Classes and Access Control	83
Brief Review	84
Defining a Class	84
Defining an Object	85
Referencing Fields or Methods	85
Defining Methods	86
Adding a Result type	86
Providing an Argument List	87
Method Types	87
Query Methods	88
Mutation Methods	89
Helper Methods	90
Controlling Access	91
Modifiers for Classes	91

Modifiers for Methods and Data Fields	91
Other Modifiers	92
Creating Objects	94
Initializing an Object	94
Accessing Objects	96
toString Method	96
Equality Method	97
HashCode Method	97
The this Reference	98
Defining a Constant Object	99
Class Variables and Methods	100
General Guidelines	101
Abstract Classes and Interfaces	103
Reusing a Class	104
Extending a Class	104
Subclass Constructors	106
Constructor Chaining	107
Abstract Classes	108
Interfaces	110
Overriding a Polymorphic Method	113
Final Classes	127