Preface

This concise textbook has been primarily designed for undergraduate students as a very first course in programming. The book requires no prior knowledge of programming nor algorithms. It provides a gentle introduction to these topics. The contents of this book have been organized into ten chapters split over *two parts*, as follows:

– The first part is concerned with *getting ready* to program basic tasks using the modern language Java$^{\text{TM}}$. The fundamental notions of variables, expressions, assignments with type checking are first explained. We present the conditional and loop statements that allow programmers to control the instruction work flows. The concepts of functions with pass-by-value arguments and recursion are explained. We proceed by presenting arrays and data encapsulation using objects, and insist on the notion of references for the latter.

– The second part of the book focuses on *data-structures* and *algorithms*. We first describe the fundamental sequential and bisection search techniques, and analyze their respective efficiency using complexity analysis. Since the effective bisection search requires sorted data, we then explain basic iterative and recursive sorting algorithms. We follow by explaining linked lists and describe common insertion/deletion/merge operations on them. We then introduce the concept of abstract data-structures (illustrating them with queues and stacks) and explain how to program them in Java using the object-oriented style methods. Finally, the last chapter is an introduction to more evolved algorithmic tasks that tackle combinatorial optimization problems.

The goal of this book is *two-fold*: Namely, during the first part, novice programmers progressively learn the basic concepts underlying most imperative programming languages using Java. The second part then introduces fresh pro-

grammers with the very basic principles of thinking the algorithmic way, and explain how to turn these algorithms into programs using the programming concepts of Java. The book progressively conveys to the reader that "programming" is in fact a complex task that consists of modeling a given problem, designing algorithms and purposely structuring its data for solving the problem, coding the algorithm into a program, and finally, testing the program.
Each chapter of the book concludes with a set of exercises that lets students practice notions covered by the chapter. The third part of the book consists of an overall exam that allows readers to evaluate their assimilation level. A solution is provided. Exercises and sections that are recommended to be skimmed through in a first reading are indicated using the mark **.
Additional materials, including all Java source codes for each chapter, are available at the following book web page:

`http://www.lix.polytechnique.fr/Labo/Frank.Nielsen/JavaProgramming/`

Preface for Instructors

The Java programming curriculum (called INF311) from which this book has been prepared has been taught at École Polytechnique (Palaiseau, France) for many years. Every year about 250 students enroll into the curriculum. For most of them, it is their first experience with the Java programming language. Some of them have a prior programming experience using mathematical packages such as Maple™ which are interpreters. This yields an important source of confusion not only from the standpoint of the language syntax but also from the conceptual sides of using an imperative programming language. The INF311 curriculum does not assume any prior programming experience and concentrates on teaching the fundamental notions of programming an imperative object-oriented language. This book is intended as a first programming course with two objectives in mind:

– Get a firsthand experience on programming basic algorithms using basic features of Java, and
– Introduce the very first fundamental concepts underlying computer science (that is, complexity analysis, decidability, abstract data-structures, etc.).

The curriculum consists of ten lectures (each of them lasts *90 minutes*) that deal with the following topics:

Lecture 1	Variables, expressions and assignments (with an introduction to the science of computing)
Lecture 2	Conditional and loop statements
Lecture 3	Static functions and recursions
Lecture 4	Arrays
Lecture 5	Objects (data encapsulation without object methods)
Lecture 6	Rehearsal for mid-term programming exam
Lecture 7	Searching and sorting
Lecture 8	Linked lists
Lecture 9	Data-structures and object methods
Lecture 10	Combinatorial optimization algorithms

A first course in programming without hands-on experience on writing programs by oneself is simply not conceivable. That is why each lecture is followed by a two-hour programming training class to let students become familiar with the notions covered during the lectures, and experience for themselves tracking and correcting bugs.

To control the level of assimilation by students, we organize at mid-term of the curriculum a two-hour programming exam that is semi-automatically checked using scripts and Java input/output redirections. The final exam is a two-hour paper exam that focuses more on checking whether students understand the basic data-structures and algorithms. A review exam with a detailed solution is provided in Chapter 11 (page 227).

The pedagogic resources, which include slides of each lecture and recorded videos of the lectures taught in the auditorium, are available at the following web page:

`http://www.enseignement.polytechnique.fr/informatique/INF311/`

Frank Nielsen December 2008.

École Polytechnique (Palaiseau, France)
Sony Computer Science Laboratories, Inc. (Tokyo, Japan)

Acknowledgments

It is my great pleasure to acknowledge my colleagues, Professors François Morain and Robert Cori at École Polytechnique (France), who taught this introductory course. I have greatly benefited from their course materials but more importantly from their feedback and advice. I express my deepest thanks to Philippe Chassignet and the full team of teaching assistants for their shared passion of letting students get their first programs to compile and work; Many thanks to Bogdan Cautis, Guillaume Chapuy, Philippe Chassignet, Etienne Duris, Luca de Féo, Yann Hendel, Andrey Ivanov, Vincent Jost, Marc Kaplan, Gaëtan Laurent, David Monniaux, Giacomo Nannicini, Sylvain Pradalier, Stéphane Redon Maria Naya Plasencia, Andrea Roeck, David Savourey, and Olivier Serre.

This book became possible thanks to the warm encouragement and support from Sony Computer Science Laboratories, Inc. I express my gratitude to Dr. Mario Tokoro and Dr. Hiroaki Kitano, as well as all other members of Sony Computer Science Laboratories, Inc.

I thank Ian Mackie for asking me to write this undergraduate textbook for the Undergraduate Topics in Computer Science series on behalf of Springer-Verlag.

I apologize for any (involuntary) name omissions and remaining errors.

Finally my deepest thanks and love go to my family for their everlasting support.

http://www.springer.com/978-1-84882-338-9

A Concise and Practical Introduction to Programming
Algorithms in Java
Nielsen, F.
2009, XXVIII, 252 p. 27 illus., Softcover
ISBN: 978-1-84882-338-9