# PREFACE

## WHAT IS THE PRIMARY VALUE PROPOSITION FOR THIS BOOK?

This book contains a fast-paced introduction to as much relevant information about data structures in `Java` as possible that can be reasonably included in a book of this size.

## THE TARGET AUDIENCE

This book is intended primarily for people who have some exposure to Java and are interested in learning about data structures. This book is also intended to reach an international audience of readers with highly diverse backgrounds in various age groups. While many readers know how to read English, their native spoken language is not English (which could be their second, third, or even fourth language). Consequently, this book uses standard English rather than colloquial expressions that might be confusing to those readers. As such, this book endeavors to provide a comfortable and meaningful learning experience for the intended readers.

## WHAT WILL I LEARN FROM THIS BOOK?

The first chapter contains a quick introduction to Java, along with some Java code samples to check for leap years, find divisors of a number, and working with arrays of strings. The second chapter introduces recursion and contains code samples to check if a positive number is prime, to find the prime divisors of a positive integer, to calculate the GCD (greatest common divisor) and LCM (lowest common multiple) of a pair of positive integers.

The third chapter contains Java code samples involving strings and arrays, such as finding binary substrings of a number, checking if strings contain unique characters, counting bits in a range of numbers, and how to compute XOR without using the XOR function.

Chapters 4 through 6 contain Java code samples involving search algorithms, concepts in linked lists, and tasks involving linked lists. Finally, Chapter 7 discusses data structures called queues and stacks, along with some Java code samples.

## DO I NEED TO LEARN THE THEORY PORTIONS OF THIS BOOK?

Once again, the answer depends on the extent to which you plan to work with data structures in Java. In general, you will probably need to learn many (most?) of the topics that you encounter in this book if you are plan to advance beyond a beginner-level developer in application development in Java.

## GETTING THE MOST FROM THIS BOOK

Some programmers learn well from prose, others learn well from sample code (and lots of it), which means that there's no single style that can be used for everyone.

Moreover, some programmers want to run the code first, see what it does, and then return to the code to delve into the details (and others use the opposite approach).

Consequently, there are various types of code samples in this book: some are short, some are long, and other code samples "build" from earlier code samples.

## WHAT DO I NEED TO KNOW FOR THIS BOOK?

A basic knowledge of Java is the most helpful skill, and some exposure to recursion and data structures is obviously helpful. Knowledge of other programming languages can also be helpful because of the exposure to programming concepts and constructs. The less technical knowledge that you have, the more diligence will be required in order to understand the various topics that are covered.

*If you want to be sure that you can grasp the material in this book, glance through some of the code samples to get an idea of how much is familiar to you and how much is new for you.*

## DON'T THE COMPANION FILES OBVIATE THE NEED FOR THIS BOOK?

The companion files contain all the code samples to save you time and effort from the error-prone process of manually typing code into a text file. In addition, there are situations during which you might not have easy access to the companion files. Furthermore, the code samples in the book provide explanations that are not available on the companion files.

## DOES THIS BOOK CONTAIN PRODUCTION-LEVEL CODE SAMPLES?

The primary purpose of the code samples in this book is to show you how to use Java in order to solve a variety of programming tasks. Clarity has higher priority than writing more compact code that is more difficult to understand (and possibly more prone to bugs). If you decide to use any of the code in this book in a production environment, you ought to subject that code to the same rigorous analysis as the other parts of your code base.

Another detail to keep in mind is that many code samples in this book contain "commented out" code snippets (often `println()` statements) that were used during the development of the code samples. Those code snippets are intentionally included so that you can uncomment any of those code snippets if you want to see the execution path of the code samples.

## WHAT ARE THE NONTECHNICAL PREREQUISITES FOR THIS BOOK?

Although the answer to this question is marore difficult to quantify, it's especially important to have strong desire to learn about data structures, along with the motivation and discipline to read and understand the code samples.

## HOW DO I SET UP A COMMAND SHELL?

If you are a Mac user, there are three ways to do so. The first method is to use `Finder` to navigate to `Applications > Utilities` and then double click on the `Utilities` application. Next, if you already have a command shell available, you can launch a new command shell by typing the following command:

```
open /Applications/Utilities/Terminal.app
```

A second method for Mac users is to open a new command shell on a MacBook from a command shell that is already visible simply by clicking `command+n` in that command shell, and your Mac will launch another command shell.

If you are a PC user, you can install Cygwin (open source *https://cygwin.com/*) that simulates bash commands, or use another toolkit such as MKS (a commercial product). Please read the online documentation that describes the download and installation process. Note that custom aliases are not automatically set if they are defined in a file other than the main start-up file (such as .bash_login).

## COMPANION FILES

All the code samples and figures in this book may be obtained by writing to the publisher at *info@merclearning.com*.

## WHAT ARE THE "NEXT STEPS" AFTER FINISHING THIS BOOK?

The answer to this question varies widely, mainly because the answer depends heavily on your objectives. One possibility involves learning about more complex data structures and implementing them in Java. Another option is to prepare for job interviews for careers involving Java.