Java for Python Programmers

Bradley N. Miller

January 27, 2008

1 Preface

Welcome to *Java for Python Programmers*. This short ebook is an ongoing project to help Computer Science students who have had one or two semesters of Python learn the Java programming language. If you are not a part of that audience you may still find this a useful way to learn about Java. This book is written using the build on what you know philosophy. In order to help you learn Java I will start with a Python example and then implement the example in Java. Along the way we will examine the strengths, weaknesses and differences between those two languages.

This book does not attempt to replace the many good Java reference books that are available, in fact I use this in my course along with Horstman's *Core Java* volumes. Please feel free to use this book for yourself, or if it fits a class you are teaching you are welcome to use this as a resource for your own class.

I have published this article using a Creative Commons license to encourage you to use it, change it, and modify it for your own purposes. I would appreciate knowing what you think if you do use this book, and I would love to see any modifications or additions you make.

Brad Miller bmiller@luther.edu January, 2008

Contents

2 Introduction

This book assumes that you are already familiar with the Python programming language. We will use Python as a starting point for our journey into Java. We will begin by looking at a very simple Java program, just to see what the language looks like and how we get a program to run. Next, we will look at the main constructs that are common to most programming languages:

- Data Types
- Loops
- Reading user input
- Conditionals

Once we have the basics of Java behind us we will move on to look at the features of Java that are both unique and powerful.

- Classes
- Interfaces
- Collections
- Graphical User Interface Programming
- Generic Programming

Please note that this book is a work in progress. I will continue to update and post new versions.

3 Why Learn another programming Language?

Python is a nice language for beginning programming for several reasons. First the syntax is sparse, and clear. Second, the underlying model of how objects and variables work is very consistent. Third, you can write powerful and interesting programs without a lot of work. However, Python is representative of one kind of language, called a dynamic language. You might think of Python as being fairly informal. There are other languages, Like Java and C++ that are more formal.

These languages have some advantages of their own. First, is speed: For very large programs Java and C++ are going to give you the best performance. Second is their maintainability. A lot of what makes Python easy to use is that you must remember certain things. For example if you set variable `x` to reference a turtle, and forget later that `x` is a turtle but try to invoke a string method on it, you will get an error. Java and C++ protect you by forcing you to be upfront and formal about the kind of object each variable is going to refer to.

In one sense Python is representative of a whole class of languages, sometimes referred to as "scripting languages." Other languages in the same category as Python are Ruby and Perl. Java is representative of what I will call industrial strength languages. Industrial strength languages are good for projects with several people working on the project where being formal and careful about what you do may impact lots of other people. Languages in this category include C++, C, C# and Ada.

Programming languages will always change. As the field of computer science advances there will be new programming languages and you will need to learn them. It is important to learn several programming languages so that you know what to expect. There are certain features that most programming languages have in common; variables, loops, conditionals, functions. And there are some features that are unique. If you know what is common in languages that is a good place to start.