Reflection report of COP2800, Java Programming

M. Ruelas Jacobo

Miami Dade College

Abstract

This report is a reflection of the author’s learning experience throughout the class and working on the final project.

Keywords: COP2800 Java Programming Learning

Reflection report of COP2800, Java Programming

My first experience in programming had been just a few months ago when I set out to learn C++ in a prior course. Java proved to be very similar and I felt the transition over was fairly smooth. There was a lot of differences in the two programming languages that made it both challenging and interesting. The class was divided into 9 Modules (0 through 8) or sections, that aim to progressively build on skills learned.

# The course

Module 0 was the introduction to the course, it included information such as the syllabus, getting the Integrated Development Environment (IDE) software installed, and introducing oneself to the rest of the class. The main source of information for the class was the book “Starting out with Java: Early Objects” by Tony Gaddis. In his usual style he begins by giving an introduction to the computer science field and the programming language. I think he does a great job in getting the necessary foundation needed to have a “big picture” view of what happens within a computer and the role software/programming takes.

## Module 1

Module 1 consists of the initial two chapters of the book, “Introduction to Computers and Java” and “Java Fundamentals”

Introduction to Computers and Java. This chapter served to initiate students into the world of computers and how they functioned. For persons already in Information Technology, it helped remind them terminology and how computers and software operate. The main takeaways for me were that “Computer programming is both an art and a science”, the inner workings of a CPU and how memory works at storing and retrieving information. The rest of the chapter is devoted to acquainting students to the Java language, object oriented programming and the process of programming.

Java Fundamentals. This is where things get real, and we get right into creating a program. We are given an overview of the different parts of a Java program, what the basic syntax means, and how to do basic things like display text, gather input, store information in variables and do some arithmetic operations.

## Module 2

Decision Structures. After learning how to store information, our next objective was to learn how to use the information to make decisions. In my opinion this chapter is what drives the rest of the programming and is essential for any prospective programmer to thoroughly understand. One of the hurdles I faced in this chapter and throughout the course was working with string objects as unlike C++, you must use the String method to accomplish comparisons. I also found the declaring of variables to be somewhat challenging as the IDE I used to code would give errors if a variable was not initialized, even if it would be initialized somewhere in the code before its use.

Loops and Files. Loops is another major foundational piece; they provide a way to automate repetitive tasks. The piece that interested me the most in this chapter was learning how to work with files. I found this section to be pretty exciting and fun.

## Module 3

A First Look at Classes and Objects. Just as students thought they were getting a good grasp on Java, classes and objects are brought into the mix. This brings us out of our nice and neat box were we can only color inside the lines to a world of variety and ambiguity. The reality of programming quickly sets in were students quickly learn that any lack of planning or break in though can cause a lot of heartache. Objects and classes add a layer of versatility and complexity that enables the creation of large and complex programs and the ability to reuse code by extending classes and overloading methods/constructs.

A Second Look and Classes and Objects. Building on the foundation of the prior chapter, we learn more about classes and objects in regards to fine tuning and memory management. Static class members and fields allow better memory management by having one copy used for all instances of a class and Garbage collection removes unneeded items from memory. We also learn about working with methods, sending and receiving information and using methods to handle data/objects.

## Module 4

Arrays and the ArrayList Class. Arrays and Arraylist’s provide a way to handle sets of data and create relationships between data. By using subscripts and the “for” loop we can quickly access and store information. While working with arrays I envisioned the creation of an excel sheet and imagined the world of databases being composed of large multidimensional arrays.

Text Processing and Wrapper Classess. At first the wrapper classes seemed somewhat confusing, but after learning how they provide versatility to working with primitive data types, I found them to be amazing! These classes allowed programmers to easily convert and test data without having to write their own methods to do so.

## Module 5

Inheritance and Polymorphism. It makes sense to have this chapter be its own module. To me, this chapter was about re-using code by extending, overriding and or abstraction. I envisioned this as the moment when “Neo” from the matrix learned that he could change the world at will.

## Module 6

Exceptions and Advanced File/IO. Exceptions were pretty easy to follow and understand. They allow programmers to allow their programs to handle foreseeable errors that may occur due to programming or user error. The real interesting part of this module for me was the advanced file operations.

## Module 7

GUI Applications and JavaFX. Creating a GUI was like learning a whole new language and it seemed to me like it could be a whole class of its own. This made me appreciate programs even more, as there is so much work that goes in to creating them. Not only do you have to build the display, but also what happens when a user makes a selection, closes the window or pushes a button.

## Module 8

Recursion, Applets and More…. Recursion allows a programmer to create a way for code to repeat itself. Similar to a “for” loop but instead using a method. This opens the door for using methods to handle algorithms and repetitive tasks while limiting the amount of code that needs to be written. Applets are extremely useful as it allows your programs to run inside a web browser.

## Final Project

[Heading 3]. The final project was challenging. I reference the book and using the many examples of code I found I had all the tools needed to complete it. The instructions from the teacher were very good which allowed me to create a plan for success. The most challenging portion of the project was creating a GUI interface.

# Conclusion

I highly recommend programming to everyone. Java was a very fun language to learn and the course helped me fine tune my knowledge and skills. As always there is much more to learn but I feel very good that the skills and knowledge I attained will aid me well in the future.

References

Gaddis, T. (2015). *Starting Out with Java* Pearson