



Ross Buffington

The Objective-C Programming Language

Introduction

There are many programming languages present in today's technological world. These languages range from being widely popular to having only a handful of implements. Every language is built for a specific purpose and audience and there is no existing language that is suitable for every purpose.¹ For our Programming Languages class, I have chosen to study Objective-C. I feel this language has a wide variety of uses, many of which I find highly interesting, and the language presents a good opportunity for me to grow as an object-oriented programmer.

History

Objective-C was created in 1987 (the year of my birth) by Brad Cox and Andrew Novobilsky.² It is based on the C programming language and adds objects and messaging in the style of Smalltalk (another influence). The language was one of the first to adequately bridge the gap between human thinking and computer thinking.³

Later in the evolution of Objective-C, Apple Inc. adopted it and began to widely implement the language. Objective-C became the foundation of Apple's Cocoa framework, the layer that sits on top of UNIX in of the popular operating system, Mac OS X.

Apple has taken over development of the Objective-C programming language and recently released version 2.0. One of the most important features in this new version is automatic garbage collection, removing the task of memory management from the programmer's shoulders and allowing the OS to manage memory instead.⁴ By Apple's development standards, native OS X applications are to implement as much of the Cocoa framework as possible, so Objective-C is a major aspect of software development for the platform. Apple's iPhone 2.0 extensively implements Objective-C in it's Cocoa-Touch framework. Therefore, it is imperative for any programmer who wants to dive into the iPhone application development market to be a master of Objective-C.⁵

While Objective-C continues to progress, it has also had influence on other languages. Both Java and C# have adopted the idea of protocols (Java calls them "Interfaces") as a way to achieve multiple inheritance. Objective-C++ has also been birthed by this language.⁶

The future of Objective-C looks good. Since the language is at the core of an operating system that is rapidly abducting Windows users, as well as at the core of a revolutionary mobile productivity solution (the iPhone), I would say the chances of this language becoming obsolete, any time soon, are 'nil' (Objective-C's word for *null*).⁷

¹ Crowell, Joyce Blair. "Programming Language Lecture: Chapter One." Belmont University. Sept. 3, 2008.

² <http://virtualschool.edu/objectivec/>

³ Kay, Alan. "The Early History of Smalltalk".

⁴ <http://developer.apple.com/documentation/Cocoa/index.html>

⁵ <http://developer.apple.com/iphone/library/navigation/Frameworks/CocoaTouch/index.html>

⁶ http://developer.apple.com/documentation/Cocoa/Conceptual/ObjectiveC/Introduction/chapter_1_section_1.html

⁷ <http://developer.apple.com/iphone/>

Purpose

Objective-C is an object-oriented programming language. Therefore, a programmer can potentially develop whatever they want with the language. Its roots in C give the programmer access to low-level operating system functionality while its object-oriented design and related GUI frameworks give the programmer complete control in higher-level programming solutions.

While Objective-C is extremely powerful, there are some limitations. Mac OS X and the iPhone are by far the largest implementations of the language and because of this, it is doubtful that a Windows user would find many problems where Objective-C would be the programming language of choice. However, GCC is able to compile Objective-C so any machine capable of running the compiler is, in theory, able to wield the power of this beautiful language.⁸

I consider the iPhone to be the number-one best outlet for programming in Objective-C. In the current mainstream, there is a huge buzz about iPhone Applications and the potential to make a few bucks off of fairly easy programming projects is quite high. With the extreme focus on object-oriented design patterns, designing and coding applications for the iPhone is challenging, compelling, and just downright cool! That's why I'm going to make all of the programs for this class run on the iPhone!

Why It's My Choice

I have chosen to program in Objective-C for one main reason and several sub-reasons. Mainly, I am a member of the cult of Apple Products; I drank the Cool-Aid...about 30 gallons of it. I love the feel of the *MacBook Pro* I am using to type this paper, I love the design of the *Pages* word processor I am using, and I love my *iPhone*. As I will never be switching back to a Windows machine, and in some cases a piece of software requires a native language, I would like to learn Objective-C. Additionally, this will help me bond with my Mac.⁹

A sub-reason that I find very important is the focus on object-oriented programming that Apple and Objective-C stress. All of the documentation for the language is geared toward object-oriented programming. This fact tells me that learning this language and holding fast to its development standards will make me a better software designer. I think it is important to improve this skill, and it may even be more important than learning specific languages, because it enables Computer Science students of the USA to actually get jobs, when simple programming work is increasingly outsourced to India, China, etc. Therefore, I am thankful for the opportunity to take this class and look forward to the challenge of learning Objective-C.

⁸ <http://www.cs.indiana.edu/classes/c304/ObjCompile.html>

⁹ While I am very opinionated in this paragraph I do not mean to sound obnoxious. Mac is just my preference and I like to exaggerate.