



INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact factor: 4.295

(Volume 3, Issue 6)

Available online at www.ijariit.com

Raising Performance of iPhone using Swift Language over Other Programming Languages

Bikramjit Singh

Computer Science & Engineering
Doaba Group of College, PTU, Punjab, India
bikram4127@gmail.com

Ramanjot Kaur

Computer Science Engineering
Doaba Group of College, PTU, Punjab, India
raman0803@gmail.com

Abstract: The new programming language gives the fundamental to foil its importance with the current programming language to calculate the furtherance the new programming language gives to the developers. These types of studies can show the importance, improvements, and purpose of new programming languages. In this article, we analyze the new programming language of Apple that is swift with the existing programming language (Objective-C). We are going to describe the changes, components, and freshness to verify the words of Apple said about the swift language.

For the replacement of objective-C, Swift needs to assure that it contains a lot of new features as Apple said. Apple has sure-fire that swift will be faster than the Objective-C, as well as it contains a lot of modern language features that are necessary for today's era. In this thesis, it is justified by creating iOS application totally in swift. Developer finds that swift is faster and more efficient by Objective-C. It's proof that swift have a lot of new features and many advantages over the Objective-C and it is also easy to learn.

Swift was launched to offer replacement of Objective-C because this has a syntax which barely evolved from the time of its creation and has a big difference with other programming languages that have appeared in the modern era, because they have based on C++ syntax. For this, Swift is inspired in a new programming language like C++, C#, Go, Java, Python, Ruby, JavaScript. Then this syntax is different than the former language. The Swift's syntax is more simple because it does not use pointers and includes improvements in its data structures and in its syntax. Swift has a simple syntax which helps developers to have fewer mistakes. Swift is an object-oriented and imperative programming language as Objective-C but swift incorporates the functional programming. Some examples of this are filters, amp. Due to these facts, it is important to learn swift to check if Swift could be a programming language adapted to the new times and if it could facilitate the application development for platforms.

Keywords: Swift Language, Ruby, Python, Objective-C, C#, iOS

I. INTRODUCTION

[I] i. Introduction to Swift Language

Swift is a powerful and intuitive programming language for MACOS, iOS, watchOS, and tvOS. Writing Swift code is interactive and fun, the syntax is concise yet expressive, and Swift includes modern features developers love. Swift code is safe by design, yet also produces software that runs lightning- fast. Swift combines powerful type inference and pattern matching with a modern, lightweight syntax, allowing complex ideas to be expressed in a clear and concise manner. As a result, the code is not just easier to write, but easier to read and maintain as well. It is built with the open source LLVM compiler framework and has been included in Xcode since version 6. On platforms other than Linux, it uses the Objective-C runtime library which allows C, Objective-C, C++ and Swift code to run within one program. Swift was introduced at Apple's 2014 Worldwide Developers Conference (WWDC).

It underwent an upgrade to version 1.2 during 2014 and a more major upgrade to Swift 2 at WWDC 2015. Initially, a proprietary language, version 2.2 was made open-source software under the Apache License 2.0 on December 3, 2015, for Apple's platforms and Linux. Swift is an alternative to the Objective-C language that employs modern programming-language theory concepts and strives to present a simpler syntax. During its introduction, it was described simply as "Objective-C without the C".

[I] ii. Python

Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java.

[I] iii. Objective-C in Swift Language

Swift is designed to provide seamless compatibility with Cocoa and Objective-C. You can use Objective-C APIs in Swift, and you can use Swift APIs in Objective-C. This makes Swift an easy, convenient, and powerful tool to integrate into your development workflow. Swift allows to incorporate and use the objective-C code in the same swift program. To insert the objective-C file, you must insert Objective-C ".h" and ".m" files and import the "Bridging-Header" file. This file contains the import all the objective-C header files. This guide covers three important aspects of Swift and Objective-C compatibility that you can use to your advantage when developing Cocoa apps.

II. LITERATURE SURVEY

By Apple's Developers Team (2014) Development on Swift was begun in July 2010 by Chris Lattner, with the eventual collaboration of many other programmers at Apple. Swift took language ideas "from Objective-C, Rust, Haskell, Ruby, Python, C#, CLU, and far too many others to list". On June 2, 2014, the Apple Worldwide Developers Conference (WWDC) application became the first publicly released app written in Swift. A beta version of the programming language was released to registered Apple developers at the conference, but the company did not promise that the final version of Swift would be source code compatible with the test version. Apple planned to make source code converters available if needed for the full release.

Leif Oppermann and Jubilee Campus In almost every field of scientific inquiry, the current rate of scientific publication is greatly outpacing scientist's ability to read and assimilate the information. It has been estimated that every year, more than 4000 systematic reviews are conducted and published, each with the goal of summarizing the current state of knowledge relevant to a specific research question. On average, the amount of time required to conduct a single systematic review is at least 6 months to a year, and a considerable portion of this time is often spent on formulating the problem and identifying the relevant literature. For this reason, a large number of topics that would benefit from systematic review are waiting in the queue and many systematic reviews are out of date by the time they are published.

Pedagogy The landscape in which a language and/or tool is deployed consists of the subject matter and skills acquisition associated with taking the course, and the approach to teaching and learning programming adopted by the instructor. While, the curriculum defines what is to be taught, Pedagogy deals with the manner in which teaching and learning are managed in order to facilitate desired learning outcomes. The approach taken to structuring the learning situation influences both the expected outcomes and the techniques adopted to achieve them. There is a large body of general literature on the roles and methods teaches and students encounter in higher education.

By Christ Lattner (2015) Released in June of 2014 by Apple Swift is a statically typed language and compiled a language that uses the LLVM compiler infrastructure and the Objective-C runtime. Since Swift uses the same runtime as Objective-C the two languages can be intermixed in a single program or project, as both will compile down to native machine code. Swift can access Objective-C classes, types, functions, and variables through a "bridging header", as well as by extension C and C++ code. Similarly, Objective-C can access code written in Swift, with some exceptions. This allows Swift to work with the Cocoa and Cocoa Touch frameworks and existing Objective-C apps and libraries without rewriting the large body of code that was written for iOS devices. Swift is heavily influenced by many other languages such as Rust, Haskell, Ruby, Python, and C#, and offers many of the object-oriented and functional features found in these languages. Swift also includes a read-eval-print-loop (REPL) that can be accessed in Xcode as well as on the command line.

III. OBJECTIVES

- To calculate the speed of code execution with Swift language over Objective-C and other programming language used in the iOS mobile application.
- To measure the performance of Swift language on iOS mobile application platform.
- Finding out feature set and safety course for Swift language.

IV. METHODOLOGY

Swift and Objective-C compilers are based on the LLVM Compiler Infrastructure, and there is a single iOS SDK for both Swift and Objective-C. That's why there isn't much difference between the ways the programming languages work with the Cocoa frameworks. We decided to examine both Swift and Objective-C performance by comparing their data structures.

For that, we took Objective-C Foundation framework and Swift's native solutions. In Swift, all classes are created during compile-time. Methods cannot be added on-the-fly and all types are known before the run time. Since everything is known beforehand, a compiler can optimize code without any problem. Objective-C, on the other hand, can't optimize as effectively, because all dynamic

languages work slower than static. Swift drops the two-file requirement. Xcode and the LLVM compiler can figure out dependencies and perform incremental builds automatically in Swift 1.2. As a result, the repetitive task of separating the table of contents (header file) from the body (implementation file) is a thing of the past. Swift combines the Objective-C header (.h) and implementation files (.m) into a single code file (.swift). Objective-C's two-file system imposes additional work on programmers -- and its work that distracts programmers from the bigger picture. In Objective-C you have to manually synchronize method names and comments between files, hopefully using a standard convention, but this isn't guaranteed unless the team has rules and code reviews in place.

V. RESULT

We Perform task as run for loop 10 times to multiply 100x100 matrices together with n3 algo in every five languages. In this test swift far outperformed Objective-C by a significant margin, running on average almost 8 faster than Python and 3x faster than Objective-C, While still not quite as fast as swift's performance was on par with java. While these results are approximate they still show that it is unlikely that swift is quit as fast as a project by Apple.

Features of Swift Language from other Languages

- Closures: JavaScript
- Generics: Java
- Type inference: Haskell
- Tuples: Python
- Functions: JavaScript
- Operator Overloading: C++
- Pattern Matching: Scala
- Optional: Rust
- ARC: Objective-C
- Protocols: Java
- Read-Eval-Print-Loop: python

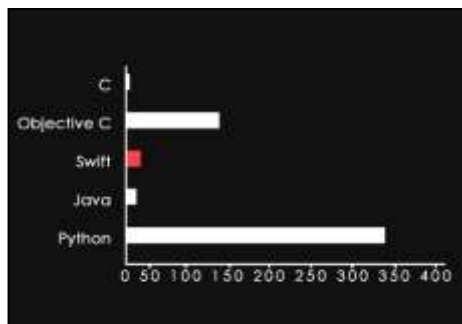


Figure1: Different Languages Time Elapsed In Seconds



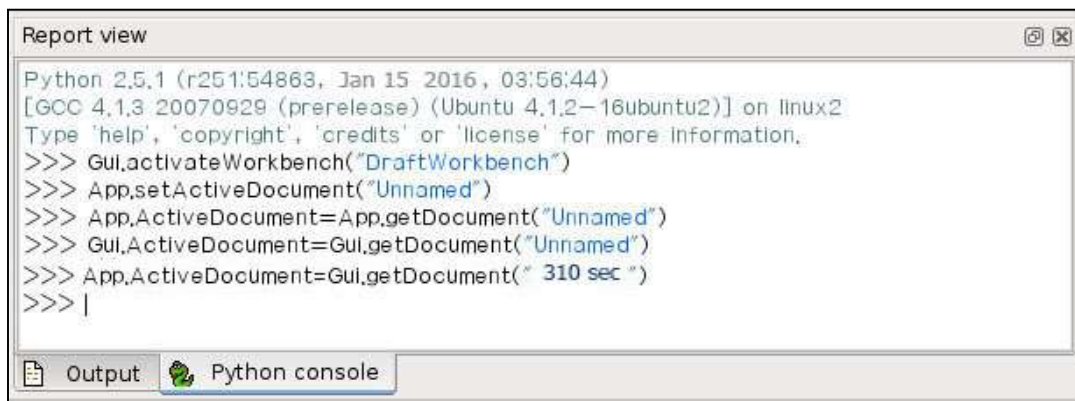
Figure 2: Swift Language Algorithm Execution Time



Figure 3: Objective-C Language Algorithm Execution Time



Figure 4 Java Language Algorithm Execution Time



```
Report view
Python 2.5.1 (r251:54863, Jan 15 2016, 03:56:44)
[GCC 4.1.3 20070929 (prerelease) (Ubuntu 4.1.2-16ubuntu2)] on linux2
Type 'help', 'copyright', 'credits' or 'license' for more information.
>>> Gui.activateWorkbench("DraftWorkbench")
>>> App.setActiveDocument("Unnamed")
>>> App.ActiveDocument=App.getDocument("Unnamed")
>>> Gui.ActiveDocument=Gui.getDocument("Unnamed")
>>> App.ActiveDocument=Gui.getDocument(" 310 sec ")
>>> |
```

Figure 5: Python Language Algorithm Execution Time

VI. CONCLUSION

Swift is clearly far safer than Objective-C and scripted languages like JavaScript. This is primarily due to the use of static typing and of optionals. It is also due to a syntax that is simple, clear, and well-designed. The use of type inference allows a high degree of type safety while putting a little burden on the programmer. Optional does put a significant burden on the programmer, and it is quite possible that many programmers will find the ways to avoid using them in the way they are most effective, so there is some uncertainty about their real impact.

Clarity, Swift is very clear, well- designed language, with many developers saying that it is “a joy to use”, and a huge improvement on Objective-C’s ugliness and clutter.

Modernity, Swift’s closure and functions, including them as “First class citizens” that can be passed around as parameters and return values, are well defined. The ability to iterate through strings, arrays, and dictionaries, the addition of structures and enumerations, the ability to add new operators and functions and to do generic programming are also substantial ways in which the new language is modern.

Swift Means for iOS Development

The reaction to Swift from the developer community has generally been very positive. Criticism has been minor. This is perhaps surprising given the programmer culture in which people have strong and often crazy opinions about everything, no matter how ill-informed. Some of the negative reactions have been predictable: Enthusiasts of scripting languages like JavaScript think that not having implicit type conversion (automatically converting a data value’s type when necessary) makes the language too inflexible. This is mostly a philosophical disagreement that has no solution. Some programmers feel that errors are inevitable and want their programs to run no matter what. Others want them to quickly fail in hopes of getting every last bug out.

REFERENCE

1. @Adamjleonard, @Thinkclay, and @Cesar Devers, “Swift Toolbox,” <http://www.swifttoolbox.io/>, 2014. [Online]. Available: <http://www.swifttoolbox.io/>. [Accessed: 17-apr-2015].
2. E. González, H. Fernández, and V. Díaz, “General Purpose Mde Tools,” *Int. J. Interact. Multimed.artif.intell.* Vol. 1, pp. 72–75, 2008.
3. E. R. Núñez-Valdez, O. Sanjuan-Martinez, C. P. G. Bustelo, J. M. C. Lovelle, And G. Infante-Hernandez, “Gade4all: Developing Multiplatform Videogames Based on Domain Specific Languages And Model Driven Engineering,” *INT. J. Interact. Multimed. Artif.Intell.* VOL. 2, NO. Regular Issue, PP. 33–42, 2013.
4. R. Gonzalez-Crespo, S. R. Aguilar, R. F. Escobar, and N. Torres, “Dynamic, Ecological, Accessible And 3d Virtual Worlds-Based Libraries Using Opensim And Sloodle Along With Mobile Location and NFC For Checking In,” *INT. J. Interact. Multimedia Artif Intell.* VOL. 1, NO. 7, PP. 63–69, 2012.
5. A. Puder and I. Yoon, “Smartphone Cross-Compilation framework for Multiplayer Online Games” in *2nd Int. Conf. On Mobile, Hybrid, and On-line Learning*, 2010 © IEEE Computer Society. DOI: 10.1109/ELML.2010.13.
6. A. Puder and P. Antebi, “Cross-Compiling Android Applications to IOS and Windows Phone 7” In *Mobile Netwappl*, 2012 © Springer Science Business Media, LLC. DOI: 10.1107/S11036-012-0374-2
7. Inderjeet Singh and Manuel Palmieri, “Comparison of Cross-Platform Mobile Development Tools”, IDT Malardalen University
8. C. Xin, “Cross-Platform Mobile Phone Game Development Environment” In *INT. Conf. On Industrial and Information Systems*, 2009 © IEEE Computer Society. DOI: 10.1109/IIS.2009.96
9. Inc. Apple. Using Swift with Cocoa and Objective-C. 2. Apple, INC., 2014.
- [10] Sultan S Al-Qahtani, Pawelpietrzynski, Luis F. Guzman, Rafikarif, and Adrien Tevoedjre. Comparing Selected Criteria of Programming Languages Java, Php, C++, Perl, Haskell, Aspectj, RUBY, COBOL, Bash Scripts and Scheme Revision 1.0-A Team CPL Group Comp6411-S10 Term Repor T. 2012.