Swift v/s Objective C

Meera Antu
Dept.Computer Science

Anu Maria Thomas
Dept.Computer Science

***Abstract*-: Apple may not have wowed consumers with the latest software offering as it s annual Worldwide Developers Conference in San Francisco this year.But it unveiled something that could have a much bigger impact than a new version of its iOs mobile software or even a new iPhone:a new programming language.**

Called Swift,it's meant to offer a faster,easier way to build software for Apple's product than its existing programming language,Objective-C.And so far,developers like what they see,saying it will be especially helpful for inexperienced coders who may have shied away from developing iPhone apps in the past.

Objective-Chas a long history at Apple.First conceived in the 1980s as an extension to the C programming language, it was licensed by next Computer-the company Steve Jobs founded after being ousted from Apple in 1985-and used for its operating system, too.

Yet objective-C can be complicated and confusing ,especially for new programmers,and c it is viewed as clunkier than some more modern programming language.Swift is simpler than Objective-C and allows users to see their code in action as they write it on their computer screens,rather than running it through a compiler before checking out.But it also compatible with Objective-C code,which means it will work with apps already in Apple's App store.And it's meant to be used for development of Mac apps,too.

Apple rolled out a beta version of Swift on Monday to members of its developer program.A final version will come out in the fall.

Introducing Swift to the crowd a Apple's annual developer conference in San Francisco on Monday,Craig Federighi, Apple's senior vice president of software engineering,said it's"like Objective-C without the bulk of C."

I. INTRODUCTION

Swift is a new programming language for iOs X aps that builds on the best of C and Objective-C,without the constraints of C compatibility.Swift adopts safe programming patterns and adds modern features to make programming easier,more flexible,and more fun.Swift's clean slate ,backed by the muture and much –loved Cocoa Touch frame work,is an opportunity to reimagine how software development works.

Swift has been years in the making.Apple laid the foundation for Swift by advancing our existing compiler,debugger,and framework infrastructure.We simplified memory management with Automatic Reference Counting(ARC).our frame work stack ,built on the solid base o foundation and Cocoa, has been modernized and standardized thought.Objective-C itself has evolved to support blocks, collectionliterals, andmodules, enabling framework adoption of modern language technologies without disruption. Thanks to this groundwork, we can now introduce a new language for the future of Apple software development.Swift feels familiar to Objectives-C developers. It adopts the readability of Objective-C's named parameters and he power of Objective-C 's dynamic object model. It provides seamless access to existing Coca framework and mix-and-match interoperability with Objective-C code.Building from this common ground,Swift introduce many new features and unifies the procedural and object-oriented portions of the language.

Swift is friendly to new programmers.It is the first industrial-quality systems programming languagethat is as expressive and enjoyable as ascripting language.It supports playgrounds,an innovative features that allows programmers to experiment with Swift code And see the results immediately,without the overhead of building and running an app.

Swift combines the best in modern language thinking with wisdom from the wider Apple engineering culture.The compiler is optimized for performance,and the language is optimized for development,without compromising on either.It's designed to scale from "hello word"to an entire operating system.All this makes Swift asound future investment for developers and for Apple.

II. FEATURES

A. Error Handling Model

An advance error handling model provides clear,expressive syntax for catching and throwing errors.it's also easy to create your own custom error types so you can describe error caseswith clear,meaningful names. The new error model was designed to work seamlessly with NSError and the Cocoa frameworks.Error handling code looks like:

```
func loadData ()throws {}
func test()
{
do
{
try loadData()
}catch{
print(error)
}
}
```

Swift 2.0 has built-in availability checking to make it easy to build the best possible app for each target OS version.The compiler will give you an error when using an API too for your minimum target OS releases.

B. Syntax Improvements

The SDKs have employed new Objective-C feature such as generic and nullablity annotation to make Swift code even cleaner and safer.

C. .Open Source

Swift's unique combination of elegance,power and safety has the opportunity to move the entire software industry forward .It is existing to imagine what we will build together.

D. Modern

Swift is the result of the latest research on programming language,combined with decades of experience building Apple platform.Named parameters brought forward from Objective –C are expressed in a clean syntax that makes APIs in Swift even easier to read and maintain.Inferred types make code cleaner and less prone to mistakes,While modules eliminate headers and provide namespaces.Memory is managed automatically.All this modern thinking results in a language that is easy and fun to use .

Swift has many other features t make code more expressive

- Closure unfied with function pointer
- Tuples and multiple return values
- Generic
- Fast and concise iteration over a range or collection
- Structure that support methods,exensons,and protocols
- Functional programming patterns,e.g.,ma and filter
- Native error handling using try/catch/throw

E. Interactive Playgrounds

Playgrounds make writing Swift code simple and fun .Type a line of code and the result appears immediately. we can then quick look the result from the side of your code,or pin that result directly below.The result view an display graphics ,list of results, or graph of a value over time. We can open the Timeline-Assistant to watch a complex view evolve and animate, great for experimenting with new UI code, or to play an animated Sprite Kit scene as we code it.When we perfected or code remaining simple.

F. Read-Eval-Print-Loop(REPL)

The LLDB debugging console in Xcode includes an interactive version of the Swift language built right in. Use Swift syntax to evaluate and interact with our running app,or write new code to see how it works in a script-like environment available from within the Xcode console or in Terminal.

G. Designed For Safety

Swift eliminates entire classes of unsafe code. Variables are always initialized before use, arrays and are checked for overflow, and memory is managed automatically. Syntax is tuned to make it easy to define your internet-for example, simple three-character keywords define a variable(var)or constant(let).To provide predictable behavior Swift triggers a runtime crash if a nil optional variable is used. This crash provides consistent behavior, which ease the bug-fixing process because it forces the programmerto fix the issue right away.The swift runtime crash will stop on the line of code where a nil optional variable has been used. This means the bug will be fixed sooner or avoided entirely in Swift code.

H. Fast And Powerful

From its earliest conception,Swift was built to be fast.Using the incredibly high performance LLVM compiler,Swift code is transformed into optimized native code that get the most obvious way to write your code also performed the best.Swift is a successor to both the C and Objective oriented features such as classes, protocols,and generic giving Cocoa and Cocoa Touch developers the performance and ,power they demand.

I. Objective-C-Inter Operability

Swift code co-exist alongside our existing Objective-C files in the same project, with full access to your Objective -C API,making it easy to adopt.

111.SWIFT VS OBJECTIVE C

A.Swift Is Easier To Read

Objective-C introduced new keyword using the @symbol.Swift can unify all the keywords and remove the numerous @symbols.Swift drops legacy conventions.Thus you no longer need semicolons to end lines or parenthesis to surround conditional expressions inside if/else statements.Another large change is that method calls do not nest inside each other resulting in bracket hell—bye-bye,[[[ ]]].Method and function calla in Swift use the industry-standard comma-separated list of parameters within parentheses.The result is acleaner,more expressive language with a simplified syntax and grammar.

Swift code more closely resembles natural English, in addition to other modern popular programming language. This readability makes it easier.

B.Swift Is Easy To Maintain

Objective-C requires programmers to maintain two code files in order to improve the build time and efficiency.Swift drops the two file requirements.Xcode and the LLVM compiler can figure out dependencies and perform increment builds automatically.Swift combines the Objectives-C header (.h)and implementation files(.m)into a single code file(.swift). Xcode and the LLVM compiler can do work behind the scenes to reduce the workload on the programmer.

C. Swift Is Safer

In Objective-C,nothing happens if you try to call a method with a pointer variable that is nil.The expression or line of code becomes a no-operation.A no-op leads to unpredictable behavior .In Swift code,can generate a compiler errors as you write bad code.This creates a short feedback loop for programmers.). To provide predictable behavior Swift triggers a runtime crash if a nil optional variable is used.This crash provides consistent behavior,which ease the bug-fixing process because it forces the programmer to fix the issue right away.The swift runtime crash will stop on the line of code where a nil optional variable has been used. This means the bug will be fixed sooner or avoided entirely in Swift code.

D. Swift Is Unified With Memory Management

Swift unifies the language in a way that Objective-C never has. In Objective –C it becomes the programmers responsibility to handle memory management when working with the core graphics APIs and other low level APIs available on iOS . The huge memory leaks that a programmer can have in Objective –C are impossible in Swift.Swift do not suffer from a garbage collector running cleaning up for unused memory, Java,Go, or C#. This is an important factor for any programming language that will be used for responsive graphics and user input, especially on a tactile device like the iPhone, Apple Watch or iPad.

E. swift requires less code

Swift reduce the amount of code that is required for repetitive statements.Objective –C, working with text string is very verbose and requires many steps to combine two pieces of information.Swift adopts features like adding two strings together with a”+”operator. Swift supports string interpolation.

F. swift is faster

Dropping legacy C conventions has greatly improved Swift under the hood.Benchmarks for Swift code performance continue to point to Apple’s dedication to improving the speed at which Swift can run app logic. Primate Labs discovered that the X-code6.3 Beta improved Swift’s performance.

G. Fewer Name Collision With Open Source Projects

One issue that has plugged Objective-c code is its lack of formal support for name spaces. Swift provides implicit namespaces that allow the same code file to exist across multiple projects without causing a build failure with swift, namespaces are based on the target that a code file belongs to. This means programmers can differentiate incorporating open source projects, frame works, and libraries into your code

H. Swift Supports Dynamic Libraries

Swift has updated its major releases iOS 8,iOS 7 to dynamic libraries. Dynamic libraries are executable chunks of code that can be linked to an app. This feature allows current Swift apps to link against newer version of the Swift language. Swift can evolve faster than iOS , which is a requirement for a modern programming language.Dynamic libraries are external to be app executable.

IV. CONCLUSION

Swift is a multi paradigm, compiled programming language created by Apple.Inc. for iOS, OS X and Watch OS development. Swift is designed to work with Apple's Cocoa and Cocoa touch frameworks and the large body of existing Objective C (Obj-C) code written for Apple products. Swift is intended to be more resilient to erroneous code ("safer") than Objective-C, and also more concise. It is built with the LLVM compiler framework included in Xcode 6, and uses the Objective-C runtime, allowing C, Objective-C, C++ and Swift code to run within a single program.[8]

V. REFERENCE

[1] *https://en.wikipedia.org/wiki/Swift_(programming_language)*
[2] https://developer.apple.com/swift/
[3] https:://teamtreehouse.com