9354A
10:00-11:00 TF

SWIFT

Domaoa, Jeane Cris
Elegado, Angel Nica
Lonogan, Mehanie Jiral
Salazar, Van Yeza Mae
Willy, Kim

The Historical Perspective

The Swift programming language was developed in June 2, 2014 by Chris Lattner with the help of Apple Inc., and it was introduced in Worldwide Developers Conference (WWDC) and it undergoes upgrading to Swift 2 in the year 2015. Swift was influenced by many programming language including C#, C, CLU, Haskell, Objective-C, Python, Ruby, and Rust. In August 2014, Swift created a new document for Swift 1.0 for building an iOS and OS X apps and in the same year, they updated Swift by adding a section about Class-Only Protocols and Extensibility Protocol that can be used to types, structures and classes of a Swift. Swift removed the Inheritance in overrides and in the same way they want to show how to override and they recreate and rewrite again about the overriding property of getters and setter methods. In October 2014, they updated Swift 1.0 to Swift 1.1 in just two months. In April 2015, they update the Swift 1.1 to Swift 1.2 because they added the Set collection and updated the Operators. In September 2015, the Swift 1.2 turns into Swift 2.0 in five months only and Swift2.1 was updated in October 2015 and that's how powerful the Swift is. Last September 12, 2017 they updated Swift 4.0 version and added a new guidelines in Protocol Composition and Protocol Composition Type that is a part in type's chapter that can inherit a superclass. Swift ranked top 10 of popular programming languages in The Importance of Being Earnest (TIOBE) index in March 2017.

Language Design Goals

Every programming languages aims to create design and features which meets the needs of users and developers. In swift, they aims to develop an available language for users from desktop apps to mobile apps and other services. One of the swift goals is the source stability. Source stability allows the recent of swift compiler to accept the older versions of swift. Another goal of swift is to make strings easier to use , faster and improvements to the standard library which they successfully did. Swift is created using modern approach to the performance, safety, and software design, coding style is good. So the swift is designed to make the developers coding style easier by creating a code and maintaining programs. Another goal is to bring API stability.

Language Paradigms

- **Multi-paradigm**

A multi-paradigm programming language is one of the paradigms of Swift that has ability to permit the user to operate the qualities of styles and constructs.

Inasmuch as no individual paradigm can solve/run the problems in producing a result.

- **Protocol Programming**

A protocol paradigm supports the object-oriented programming that can interact in unconnected object. Protocol paradigm has values and method definition that can create an objects.

The protocol is a description of:

1. The object can interpret the messages.

2. The messages can produce an argument.
3. The messages can return types of results.

- **Object-oriented programming** (**OOP**)

OOP is another Swift paradigm that captures the idea of "objects", that is for containing data or information in a specific fields, sometimes called *"attributes". A form of guidelines and steps also called as "methods".*

Object-oriented programming (OOP), allow the user to communicate various people in a computer-based and class-based program.

- **Functional Programming**

Functional programming is another type of paradigms in Swift that composed of styles that create and produces a structured and element. In can also use in arithmetic computations.

In functional programming you can also do changing in state that is not depending on the function result and it is much efficient in understanding the behavior of the program, which is another way of supporting and maintaining the process of functional programming.

- **Imperative Programming**

Imperative programming is one of the paradigms of Swift for changing the programs state that uses a statement. This paradigm contains commands to perform computer task and emphasizing the concept on operating a program. Another type of imperative programming is a procedural programming that built and compiles a file or program in more than one step.

- **Code Block**

Code block is another type paradigm of Swift that includes that source code that is combined as one and has a linguistic development. Code block allows the user in building some blocks and build another block that is also known as "block-structured programming language**".**

"In a block-structured programming language, the names of variables and other objects such as procedures which are declared in outer blocks are visible inside other inner blocks, unless they are shadowed by an object of the same name."

Language Application Areas

There are many potential areas where swift is good at. One of it was web applications. A web application is a client-server application that runs in a web browser. One of the examples of web servers that uses swift is the Perfect web app. Most developers use it to build applications and its other services. One of the needs of most developers was to write shorter codes in because that is where they are comfortable and in order to for them to be more productive and efficient. So the Perfect web app is a good thing to use. There are good features that Perfect offers one of it was "No more Double vision" or code duplication. Perfect web app is one of the products of swift that can satisfy that need. Another potential area where swift thrives is in developing mobile applications. Swift is a modern programming language and it is designed to lessen the possible problems with objective-C. Because of this, some mobile developers want to use swift. There are many mobile applications in terms of chat, games, educational apps, health, etc. that

is written in swift. One of the examples in the area of mobile applications in terms of games that are written in swift is the Flappy Bird and 2048 game. There is an application named playgrounds that uses swift to create applications. It is somehow recommended to developers to make a program.

The Environment

In setting up the Swift Programming environment, you need to have an xCode application to start your Swift coding in playground. xCode was developed by Apple that was first released in 2003. xCode is an open source for the different programming languages particularly for Swift, Python, Ruby, C,C++ and Java. xCode is a complete developer toolset which you can perform your coding. You can download the xCode IDE on its official website. You can't download the xCode unless you register. If not registered, sign up and follow the procedures. Go to the main page of the Swift (developer.apple.com/swift) and click download xCode 6 beta 4. This software requires Mac running OS X 10.9.3 or higher.

In order to install, open the file that you download and follow all the instructions. Drag the xCode 6-Beta 4 to your application window. Open xCode and accept the terms and conditions. If there is no problem, click the first option which is "getting started with a playground". Enter a name for playground and select the iOS as platform.

The Tools

Swift programming language was recently created and was introduced by Apple. According to apple, Swift will "eliminate entire classes of unsafe codes" and it is "fast and powerful" so it minimizes typing allowing the developers to be more interested and creates more modern iOS applications. Here are some swift IDE for swift programming:

1.) AppCode

AppCode was developed by JetBrains and was released recently their latest update (March 2017). This IDE supports many programming language aside from swift like C, C++ and JavaScript. One of the good features of AppCode is it automatically resolves the error in your code.

2.) xCode

xCode was developed by Apple inc. it was first released on 2003. This IDE supports many programming language aside from swift it includes C, C++ and JavaScript, python and ruby.

3.) Atom

Atom is an open-source for MacOs, linux, and windows. This IDE was developed by Github Inc. It was initially released on February 2014. Atom is Open Source and it supports several programming languages SUCH AS C, C++, Phyton, Ruby, etc.

4.) Sublime Text 3

Sublime Text's initial release was January 2008.Sublime Text 3 was written in C++, Python. The Sublime Text is one of the IDE's that most people is familiar with. Sublime Text 3 is a cross platform(Windows, macOS, and Linux). One of the good features of Sublime Text is the "Goto Anything" that allows you to navigate files, symbols, and lines.

The Libraries

- Testing - Libraries for testing/running codes and test data for operating.
- Quick – A test framework for Swift and it can be used in Objective-C.
- Nimble – A test framework for matcher.
- Fakery – A test for fake data and information.
- SwiftRandom – A test in generating random data.
- MockFive – A test framework for runtime functions.
- Documentation - Libraries for documentation files.
- Jazzy – One way to create a documentation for Swift.
- Queue - Library for task queues.
- TaskQueue – This library/section includes task Queue Class
- Dispatcher – This library/section includes task groups, queues, timers.
- HTTP - Libraries for HTTP users.
- Alamofire – An http networking library.
- 3Moya – A library with abstraction in network.
- SwiftWebSocket – A library for client that has high performance WebSocke.
- iOS Http – A library for ensuring our http inquiry.
- Nuke – A library for caching image in advanced framework.
- Taylor – A library for web server
- Swifter – A library for server engine.
- Kingfisher – A library for caching image from website and downloading from the internet
- Caching - Libraries for caching.
- HanekeSwift - A library of generic cache for iOS including images.
- Carlos – A library of cache for iOS and WatchOS
- Security - Libraries for providing security and encrypting data.
- CryptoSwift – A library for helping the crypto to upgrade the Swift programming language.
- SwiftSSL – A library for crypto utilities.
- SwiftyRSA – A library of encrypting public or private key.
- Log - Libraries for log files.

- QorumLogs — A library for logging Utility that is intended for Google Docs and Xcode.
- XCGLogger – A library for debugging a log files.
- Swell – A library for logging utility that is intended for Objective C and Swift.
- Log – A library for built-in wallpapers and API in logging tool.
- NSLogger - A library that has high performance logging utility.
- Command Line - Libraries for generating and producing application in command line
- CommandCougar – A library for creating a command line applications.
- APIs Third Party - Libraries for third party authentication
- GooglePlacesAutocomplete – A library for Google that can provide the information of iOS.

The Frameworks

- Eureka - Elegant iOS Forms in Swift.
- XLActionController - Customizable and extensible action sheet
- FlourishUI – design for modals, colors, and keypad/buttons
- SwiftColors - Use for HEX color
- FontAwesome.swift - Use to create Swift projects.
- SwiftOverlays – displaying notification and reminders
- ios-charts – A presentation of graph or object
- GaugeKit – For Apples gauges that is used to make many styles.
- LNRSimpleNotifications – Simple notification for Swift
- VideoSplash – A basis for video
- EZSwipeController- Use for page view controller like Snapchat/Tinder
- Notie – A built-in application for Swift that has text field and buttons on it.
- SwiftPasscodeLock - Swift created a passcode lock with Touch Id security
- Hue - A utility for coloring
- WobbleView – Execution of wobble effect for the user's view
- EFQRCode - A best way to reply code in Swift.
- SendIndicator - Another indicator for task
- Cupcake – Used for creating and layout UI that is part of iOS
- PinLayout – Used for faster layout and it has a syntax and readable.

Official Documentations

- Swift Official Website (Swift.org)
- Apple Developers (https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift_Programming_Language/index.html#//apple_ref/doc/uid/TP40014097-CH3-ID0)
- tutorialspoint.com (https://www.tutorialspoint.com/swift/index.htm)

Main Language Features and Constructs

1. Syntax Goodies
   - Swift provides modern programming language that makes developers easier to code and understand. Swift made coding simpler like semi-colon is optional to include in the end of statements.

```
var message = "Hello, IT students!"
```

   - String interpolation is a way to define a string value containing a mix of variables, constants, expressions and literals. Each item added into the string must be covered with parentheses {()} and must be prefixed by a backslash (\).

```
print("The message is \(message)") //The message is
Hello world
```

   - You can also add elements in arrays. Arrays can be initialized without initializing its length.

```
var berry = ["banana", "potato"]
berry += ["tomato", "watermelon"]
```

2. Functions as First-Class Objects
   - The concept is simple, either both let functions take other functions as parameters, or return it as well. By this concept, it allows for greater abstractions.
   - Swift has a syntax (->) defining for functions that not exactly looks like Haskell. Before the syntax is where the parameters located while after the syntax is the return type.

```
funct greaterThanTwo (item:Int) -> Bool {
//Where (item:Int) -> Bool , is the type of
   function return item > 2
}
```

   - In arrays, we can also filter the elements with functions. Filter functions allows array based to be filtered by a certain condition.

```
var numbers = [1, 2, 3, 4, 5]
var result = numbers.filter(greaterThanTwo)
var result = numbers.filter({(num:Int) -> Bool in
return item > 2})
// We can also assign functions and use it later.
var greaterThanTwo = {(num:Int) -> Bool in return
item > 3 }
var isTwo = {(num:Int) -> Bool in return item == 2
} var result = numbers.filter(item==2 ? isTwo :
greaterThanTwo)
```

3. Type Inference-ing
   - Swift does really know the data type of a variable or a constant based on the given value of it. The interesting thing is when it is the functions that is inferred.

```
//Without Type inference
var idiot:String = "IT Students"
```

```
//With Type inference
var smart = "CS Students"
```

4. Generics
   - This feature enables developers to write functions and types that can work with any type depending on the requirements that they define. This feature is first introduced in C# and later added by Java. By using generics, developers might not use type casting.

```
func returnMe<T> (item:T) -> T
{ return item
}
returnMe(["I", "Am An", "IT Student"]).count
```

5. Tuples
   - This feature let programmers define an ordered group of values. The values of any type in a tuple can be different.

```
let message404 = (404, "Website Not Found")
```

   - We can also decompose a tuple's contents by using constants or variables

```
let (code, message) = message404
print("The code is \(code)")
```

   - We can also decompose a tuple's contents by using an underscore (_) if only a part or some of the content is needed.

```
let (_, message) = message404
print("The code is \(message)")
```

   - There are more ways to get individual/many content of a tuple but we will not discuss any further.

Sample Program Codes

- Declaring Variables and Constants

In swift programming you must have to declare your variables first before using them just like in java. In constants we use let to declare the value but for the variable we use the terminology var.

Example:

```
let customer = "Maria"
var order = "Pasta"
```

We declared a new constant called customer and has a value of Maria. Then a variable called order has a value of pizza. The constant can never be changed and we cannot re assign it. We use var if there are values that we need to change in the program. We can also declare multiple constants and variables on a single line separated by commas.

Example:

```
var firstOrder  = "Pizza", secondOrder = "Pasta"
```

- Control Flow
  - For-in Loop

    In for-in loop we iterate a sequence of statements multiple times.

    Example:

```
var table1 = ["Athena", "Aphrodite", "Artemis", "Ares"]

for customers in table1{
     print(Good Morning"\(customers)“)
}
```

We can also use swift dictionary wherein we use the identifier key that stores a value (key,value). The key can be a string or an integer and should be unique in the dictionary you created.

Example:

```
var days = [“Monday” : 20 , “Wednesday”: 75, “Friday “: 50 ]
for(weekSales, percentOfSales) in days{
print(“\(weekSales) has \(percentOfSales)% of sales”)
}
```

  - Conditional Statements
    - if or else if statement

      Tests the simple conditions in the program and throws a few results.

      The program executes the statements when the condition is either true and false then prints the value if the condition is true. Example:

```
var heightInCentimeters = 121.1
if heightInCentimeters <=  152.4{
     print(“Sorry, You’re not qualified to join the army.”)
}
else{
     print(“Congratulations! ,  Welcome to the club.”)
}
```

    - switch statement

      The switch statement is used if it has multiple conditions. You can create many cases as you can. In declaring switch statements we can declare an enum value, Integer or a String. Then we can also test two or more variables in one case by using a comma (var1,var2).

      Example:

```
  var grades = "A"

  switch grades {
        case "A" :
                  print("Excellent (90-99)")
        case "A-" :
                  print("Outstanding (80-89)")
        case "B" , "C" :
```

```
                    print("Very Good (75-79)")
            case "D"  :
                   print("Poor (Below 75) Try again next
                                semester")
            default :
                    print("Enter a value ranges from A-D ")
        }
```

- Functions

  Functions performs a specific tasks by using an organized set of statements. Functions has a return type in input parameters and has multiple return types from a single function. The function can be executed anywhere from the application so it's called the global scope.

  Example:

```
func fullName(to firstName: String, and lastName : String)->
            (String){

            return "Hi \(firstName) \(lastName) !"
        }
            print(fullName(to: "Jack", and: "Sparrow"))
```

- Classes

  Classes are like the variables, functions and constants that the programmer can declare methods and properties on it and the classes are created in a single file while the external interfaces will automatically be created when a class is initialized. Example:

```
class student{
    var idNumber = 0
    var numberOfAbsences: Int = 0

    func getDetails() -> String{
        return "The student who has an ID number of
        \(idNumber) has \(numberOfAbsences) absences."
    }
}
var sluStudent = student()
sluStudent.idNumber = 2168930
sluStudent.numberOfAbsences = 3
sluStudent.getDetails()
```

- Inheritance

  Inheritance inherits characteristics, methods, properties from another class. The classes would be subdivided into two types: sub class – inherits from another class, super class – contains the characteristics where subclasses inherit. Example:

```
class scisStudent{
var idNumber = 0
```

```
var course = "default"

       func studentDetails() -> String {
       return "This student’s ID no. \(idNumber) is
                     enrolled \(course)."
       }
}
class IT : scisStudent{
       func  programmers() -> String{
       return "IT students are idiots"
       }
}
class CS : scisStudent{
       func developers() -> String {
              return "CS students are smart"
       }
}
var sluIT = IT()
sluIT.idNumber = 2168137
sluIT.course="IT"
sluIT.programmers()
sluIT.studentDetails()

var sluCS = CS()
sluCS.idNumber = 2165184
sluCS.course="CS"
sluCS.developers()
sluCS.studentDetails()
```

References

Apple swift Programming Language Tools Now Live. (n.d.) Retrieve September 5, 207, from https://www.technobuffalo.com/204/06/02/apple-swift-programming-language-tools-now-live/

Awesome-swift. (n.d.). Retrieved September 16, 2017, from https://www.diycode.cc/projects/PhilJay/awesome-swift

Block (programming). (2017, August 21). Retrieved September 16, 2017, from https://en.wikipedia.org/wiki/Block_(programming)

Functional programming. (2017, September 07). Retrieved September 16, 2017, from https://en.wikipedia.org/wiki/Functional_programming

Imperative programming. (2017, September 15). Retrieved September 16, 2017, from https://en.wikipedia.org/wiki/Imperative_programming

M.(2016,August 16). 21 Amazing Open Source iOS Apps Written in Swift-Mybridge for Professionals. Retrieved September 15,2017, from https://medium.mybridge.co/21-amazing-open-source-ios-apps-written-in-swift5e835afee98e

More than objective-C IDE (n.d.). Retrieved September 5,207, from http://blog.jetbrains.com/objc/2014/02/more-than-objective-c-IDE/

Object-oriented programming. (2017, September 15). Retrieved September 16, 2017, from https://en.wikipedia.org/wiki/Object-oriented_programming

Point, T. (2017, August 15). Swift Constants. Retrieved September 17, 2017, from https://www.tutorialspoint.com/swift/swift_constants.htm

Point, T. (2017, August 15). Swift Dictionaries. Retrieved September 16, 2017, from https://www.tutorialspoint.com/swift/swift_dictionaries.htm

Point, T. (2017, August 15). Swift Inheritance. Retrieved September 17, 2017, from https://www.tutorialspoint.com/swift/swift_inheritance.htm

Point, T. (2017, August 15). Swift Variables. Retrieved September 17, 2017, from https://www.tutorialspoint.com/swift/swift_variables.htm

Programming paradigm. (2017, September 06). Retrieved September 16, 2017, from https://en.wikipedia.org/wiki/Programming_paradigm#Multi-paradigm

Strings and Text. (n.d.). Retrieved September 17, 2017, from https://developer.apple.com/documentation/swift/strings_and_text

Swift (programming language). (2017, September 15). Retrieved September 16, 2017, from https://en.wikipedia.org/wiki/Swift_(programming_language)

W.(2017,August 02). Retrieved September 17, 2017,from https://github.com/Wolg/awesome-swift#testing