

**Mobile Application Development (MAD2009MA)**

**MAD400-IOS Development**

**Assignment 2--18% Marks**

**Prepared By: Submitted to:**

**Sudha Rani Macherla Prof. Rajesh Khanna**

**Student Id: A00117886**

**Sandeep Lavanam**

**Student Id: A00115672**

**Date: 21-03-2021**

**Table Of Content**

|  |  |
| --- | --- |
| Number | Content |
| 1 | Introduction of Swift |
| 1.1 | Modern |
| 1.2 | Designed for Safety |
| 1.3 | Fast and Powerful |
| 1.4 | Great First Language |
| 1.5 | Source and Binary Compatibility |
| 1.6 | Open Source |
| 1.7 | Playgrounds and Read-Eval-Print-Loop (REPL) |
| 1.8 | Package manager |
| 2 | Objective-C Interoperability |
| 3 | Test Flight |
| 4 | Apple Inc., |
| 5 | Steve Jobs |

1. **Introduction of Swift**

Swift could likewise be a substitution programming language for iOS, macOS, watchOS, and tvOS application improvement. In any case, numerous pieces of Swift are visiting be recognizable from your experience of creating in C and Objective-C.

Swift gives its own adaptations of all central C and Objective-C sorts, including Int for numbers, Double and Float for drifting point esteems Bool for Boolean qualities and String for printed information. Swift additionally gives incredible variants of the three essential assortment types, Array, Set, and Dictionary, as depicted in Collection Types.

Swift is a sort of safe language, which implies the language causes you to be clear about the kinds of qualities your code can work with. In the event, that piece of your code requires a String, type security keeps you from passing it an Int accidentally. Similarly, type well being keeps you from unintentionally passing a discretionary String to a piece of code that requires a non-discretionary String. Type security causes you to catch and fix blunders as right on time as conceivable in the advancement cycle.

**1.1 Modern**

Swift is the consequence of the most recent examination on programming dialects, joined with many years of involvement building Apple stages. Named boundaries are communicated in a perfect sentence structure that makes APIs in Swift considerably simpler to peruse and keep up. Shockingly better, you don't have to type semi-colons. Derived sorts make code cleaner and less inclined to botches, while modules dispose of headers and give namespaces. To best help global dialects and emoticon, Strings are Unicode-right and utilize an UTF-8 based encoding to upgrade execution for a wide assortment of utilization cases. Memory is overseen consequently utilizing tight, deterministic reference tallying, downplaying memory utilization without the overhead of trash assortment.

struct Player {

var name: String

var highScore: Int = 0

var history: [Int] = []

init(\_ name: String) {

self.name = name

}

}

var player = Player("Tomas")

Declare new types with modern, straightforward syntax. Provide default values for instance properties and define custom initializers.

extension Player {

mutating func updateScore(\_ newScore: Int) {

history.append(newScore)

if highScore < newScore {

print("\(newScore)! A new high score for \(name)! )

highScore = newScore

}

}

}

player.updateScore(50)// Prints "50! A new high score for Tomas!

Add functionality to existing types using extensions, and cut down on boilerplate with custom string interpolations.

These ground breaking ideas bring about a language that is fun and simple to utilize.

Quick has numerous different highlights to make your code more expressive:

Generics that are incredible and easy to utilize

Convention expansions that make composing nonexclusive code much simpler

Top of the line capacities and a lightweight conclusion linguistic structure

Quick and compact cycle over a reach or assortment

Tuples and various bring values back

Structs that help techniques, augmentations, and conventions

Enums can have payloads and backing design coordinating

Utilitarian programming designs, e.g., guide and channel

Local blunder dealing with utilizing try/catch/throw.

**1.2 Designed for Safety**

Swift wipes out whole classes of hazardous code. Factors are constantly introduced before use, exhibits and whole numbers are checked for flood, memory is naturally overseen, and authorization of elite admittance to memory makes preparations for some programming botches. The linguistic structure is tuned to make it simple to characterize your purpose — for instance, straightforward three-character catchphrases characterize a variable ( var ) or consistent ( let ). What's more, Swift intensely use esteem types, particularly for usually utilized sorts like Arrays and Dictionaries. This implies that when you make a duplicate of something with that sort, you realize it will not be adjusted somewhere else.

Another well being highlight is that as a matter of course Swift articles can never be nil. Indeed, the Swift compiler will prevent you from attempting to make or utilize a nil object with an accumulate time blunder. This makes composing code a lot of cleaner and more secure and forestalls a gigantic class of runtime crashes in your applications. In any case, there are situations where nil is substantial and suitable. For these circumstances, Swift has a creative component known as optionals. A discretionary may contain nil, however Swift grammar constrains you to securely manage it utilizing the? language structure to show to the compiler you comprehend the conduct and will deal with it securely.

extension Collection where Element == Player {

// Returns the highest score of all the players,

// or `nil` if the collection is empty.

func highestScoringPlayer() -> Player? {

return self.max(by: { $0.highScore < $1.highScore })

}

}

Use optionals when you might have an instance to return from a function, or you might not.

if let bestPlayer = players.highestScoringPlayer() {

recordHolder = """

The record holder is \(bestPlayer.name),\

with a high score of \(bestPlayer.highScore)!

"""

} else {

recordHolder = "No games have been played yet.")

}print(recordHolder)// The record holder is Erin, with a high score of 271!

let highestScore = players.highestScoringPlayer()?.highScore ?? 0// highestScore == 271

Features such as optional binding, optional chaining, and nil coalescing let you work safely and efficiently with optional values.

**1.3 Fast and Powerful**

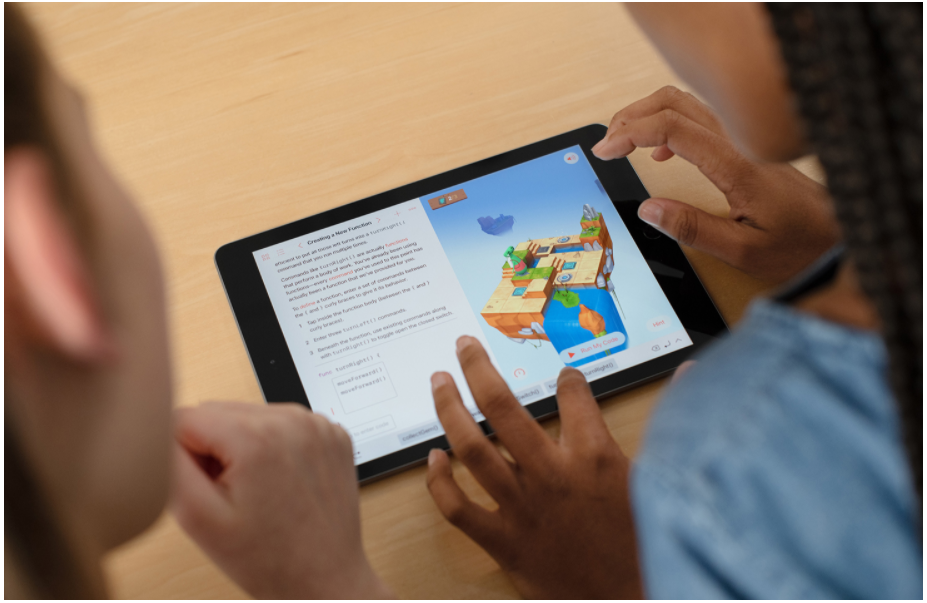
From its most punctual origination, Swift was worked to be quick. Utilizing the inconceivably superior LLVM compiler innovation, Swift code is changed into advanced local code that benefits from current equipment. The grammar and standard library have likewise been tuned to make the clearest approach to compose your code additionally play out the best whether it runs in the watch on your wrist or across a group of workers.

Quick is a replacement to both the C and Objective-C dialects. It incorporates low-level natives like sorts, stream control, and administrators. It additionally gives object-arranged highlights like classes, conventions, and generics, giving Cocoa and Cocoa Touch designers the presentation and force they request.

**1.4 Great First Language**

Swift can open ways to the universe of coding. Indeed, it was intended to be anybody's first programming language, regardless of whether you're as yet in school or investigating new profession ways. For instructors, Apple made a free educational plan to show Swift both all through the homeroom. First-time coders can download Swift Playgrounds—an application for iPad that makes beginning with Swift code intuitive and fun.

Trying application designers can get to free courses to figure out how to assemble their first applications in Xcode. Furthermore, Apple Stores around the planet have Today at Apple Coding and Apps meetings where you can get active involvement in Swift code.



**1.5 Source and Binary Compatability**

With Swift 5, you don't need to change any of your Swift 4 code to utilize the new form of the compiler. All things being equal, you can begin utilizing the new compiler and relocate at your own speed, exploiting the new Swift 5 highlights, each module in turn. What's more, Swift 5 presently presents parallel similarity for applications. That implies you at this point don't have to remember Swift libraries for applications that target current and future OS discharges on the grounds that the Swift libraries will be remembered for each OS discharge going ahead. Your applications will use the most recent rendition of the library in the OS, and your code will keep on running without recompiling. This makes building up your application more straightforward as well as lessens the size of your application and its dispatch time.

**1.6 Open Source**

Swift is created in the open at Swift.org, with source code, a bug tracker, discussions, and ordinary improvement fabricates accessible for everybody. This expansive local area of designers, both inside Apple just as many external supporters, cooperate to make Swift significantly seriously astounding. There is a significantly more extensive scope of web journals, webcasts, gatherings and meetups where engineers locally share their encounters of how to understand Swift's incredible potential.

Cross-Platform

Swift as of now upholds all Apple stages and Linux, with local area individuals effectively attempting to port to considerably more stages. With SourceKit-LSP, the local area is likewise attempting to coordinate Swift help into a wide assortment of engineer apparatuses. We're eager to see more manners by which Swift makes programming more secure and quicker, while additionally making programming more fun.

Swift for Server

While Swift forces numerous new applications on Apple stages, it's likewise being utilized for another class of present day worker applications. Quick is ideal for use in worker applications that need runtime wellbeing, assembled execution, and a little memory impression. To guide the bearing of Swift for creating and conveying worker applications, the local area shaped the Swift Server workgroup. The primary result of this exertion was SwiftNIO, a cross-stage offbeat occasion driven organization application structure for elite convention workers and customers. It fills in as the establishment for building extra worker arranged instruments and advances, including logging, measurements, and data set drivers which are all in dynamic turn of events.

**1.7 Playgrounds and Read-Eval-Print-Loop (REPL)**

Similar to Swift Playgrounds for iPad, jungle gyms in Xcode make composing Swift code unfathomably straightforward and fun. Type a line of code and the outcome shows up right away. You can then Quick Look the outcome from the side of your code, or pin that outcome straightforwardly underneath. The subsequent view can show illustrations, arrangements of results, or diagrams of worth over the long haul. You can open the Timeline Assistant to watch a perplexing perspective develop and vivify, extraordinary for exploring different avenues regarding new UI code, or to play an enlivened SpriteKit scene as you code it. At the point when you've culminated your code in the jungle gym, just move that code into your venture. Quick is likewise intelligent when you use it in Terminal or inside Xcode's LLDB investigating console. Utilize Swift punctuation to assess and associate with your running application, or compose new code to perceive how it functions in a content-like climate.

**1.8 Package Manager**

Swift Package Manager is a solitary cross-stage instrument for building, running, testing and bundling your Swift libraries and executables. Swift bundles are the most ideal approach to disperse libraries and source code to the Swift people group. Arrangement of bundles is written in Swift itself, making it simple to design targets, announce items and oversee bundle conditions. New to Swift 5, the quick run order presently remembers the capacity to import libraries for a REPL without expecting to fabricate an executable. Swift Package Manager itself is really worked with Swift and remembered for the Swift open source project as a bundle.

**2 Objective-C Interoperability**

You can make a completely new application with Swift today, or start utilizing Swift code to carry out new highlights and usefulness in your application. Quick code coincides close by your current Objective-C documents in a similar task, with full admittance to your Objective-C API, making it simple to embrace.

In Objective C, the file where the declaration of class is done is called the interface file and the file where the class is defined is called the implementation file.

A simple interface file MyClass.h would look like the following −

@interface MyClass:NSObject {

// class variable declared here}

// class properties declared here// class methods and instance methods declared here@end

The implementation file MyClass.m would be as follows −

@implementation MyClass

// class methods defined here@end

Object Creation

Object creation is done as follows −

MyClass \*objectName = [[MyClass alloc]init] ;

## Important Data Types in Objective C

|  |  |
| --- | --- |
| **Sr.No.** | **Data Type** |
| 1 | **NSString**  It is used for representing a string. |
| 2 | **CGfloat**  It is used for representing a floating point value (normal float is also allowed but it's better to use CGfloat). |
| 3 | **NSInteger**  It is used for representing integer. |
| 4 | **BOOL**  It is used for representing Boolean (YES or NO are BOOL types allowed). |

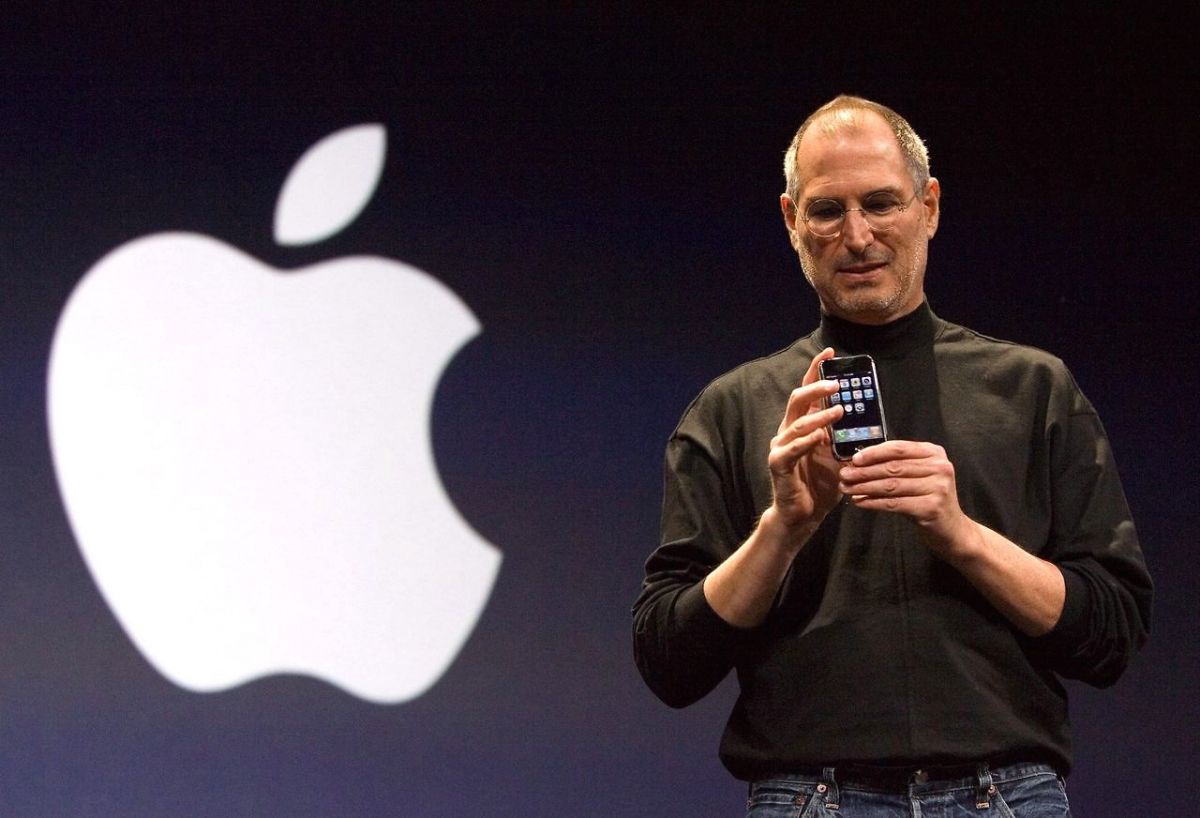
**3 TestFlight**

To exploit TestFlight, you'll need to transfer in any event one beta form of your application to App Store Connect, and welcome analyzers utilizing their email addresses or by sharing a public connection. Analyzers can begin by tolerating your email greeting or following the public connection. To introduce your application and give input, analyzers will utilize the TestFlight application for iPhone, iPad, iPod contact, Apple Watch, and Apple TV. Analyzers can turn on programmed refreshes in the TestFlight application to guarantee they're continually trying the most recent accessible form.

1. **Apple Inc.,**

iPod caused Steve Jobs to understand that Apple could turn into the best shopper gadgets organization on earth. Around 2003, he began a mysterious venture to build up a tablet. In any case, in 2004-2005, he understood that the innovation that this gathering had created, including a progressive touch-screen innovation, could be utilized in a telephone instead of a tablet. Following two additional long periods of improvement, including a brutal inner rivalry to demonstrate that it was feasible to make Mac OS X sudden spike in demand for the telephone, iPhone was presented at Macworld on January 9, 2007. This feature is frequently viewed as the best and generally paramount of all of Steve Jobs' vocation.

It is difficult to overestimate the effect of the iPhone App Store, which introduced another time in versatile programming. A great many engineers began composing applications for the iPhone stage, which turned into an upper hand for Apple that no other organization has had the option to find right up 'til the present time. Apple gladly flaunted this rich decision of programming in its TV advertisement crusade 'There's an application for that that ran for more than two years.



**5 Steve Jobs**

**Early life**

Steven Jobs was conceived February 24, 1955, in San Francisco, California, and was embraced by Paul and Clara Jobs. He grew up with one sister, Patty. Paul Jobs was a mechanic and fixed vehicles as a side interest. Occupations recollects his dad as being talented at working with his hands.

In 1961 the family moved to Mountain View, California. This region, only south of Palo Alto, California, was turning into a middle for hardware. Gadgets structure the essential components of gadgets like radios, TVs, sound systems, and PCs. Around then individuals began to allude to the region as "Silicon Valley." This is on the grounds that a substance called silicon is utilized in the assembling of electronic parts.

As a kid, Jobs favored getting things done without help from anyone else. He swam seriously however was not intrigued by group activities or other gathering exercises. He showed an early interest in hardware and gadgetry. He invested a ton of energy working in the carport workshop of a neighbor who worked at Hewlett-Packard, a gadgets producer.

Occupations additionally took on the Hewlett-Packard Explorer Club. There he saw engineers exhibit new items, and he saw his first PC at twelve years old. He was extremely intrigued and realized immediately that he needed to work with PCs.

While in secondary school Jobs went to addresses at the Hewlett-Packard plant. On one event he intensely asked William Hewlett (1931–2001), the president, for certain parts he expected to finish a class project. Hewlett was so dazzled he gave Jobs the parts and offered him a late spring temporary job at Hewlett-Packard.

**Mac and the PC time**

Occupations had acknowledged there was an enormous hole in the PC market. Around then practically all PCs were centralized servers. They were huge to such an extent that one could occupy a room thus exorbitant that people couldn't bear to get them. Advances in hardware, in any case, implied that PC segments were getting more modest and the force of the PC was expanding.

Occupations and Wozniak updated their PC, by offering them to singular clients. The Apple II went to advertise in 1977, with great first-year deals of $2.7 million. The organization's business developed to $200 million within three years. This was quite possibly the most marvelous instance of corporate development in U.S. history. Occupations and Wozniak had opened a completely new market—PCs. PCs started an altogether better approach for preparing data.

By 1980 the PC time was well in progress. Apple was consistently compelled to improve its items to stay ahead, as more contenders entered the commercial center. Apple presented the Apple III, yet the new model endured specialized and advertising issues. It was removed from the market and was subsequently improved and once again introduced.

Occupations kept on being the showcasing power behind Apple. Right off the bat in 1983, he uncovered the Lisa. It was intended for individuals having insignificant PC experience. It didn't sell well, be that as it may, in light of the fact that it was more costly than PCs sold by contenders. Apple's greatest rival was International Business Machines (IBM). By 1983 it was assessed that Apple had lost a portion of its piece of the pie (part of an industry's deals that a particular organization needs) to IBM.