**I joined EYENexo on Jan 2019 until Aug 2019, the company is a innovative ophthalmology healthcare company, during my internship, my team produced accurate eye examination software on smart phone, our application can do the test without any attachment, My strengths is to using swift SDKs and other third part SDKs, I graduated from Northeastern University by Dec 2019 with a Master of science in information system, I am trying to find a new graduate position in ios Development to improve my coding skill as well as other technology skills.**

**1- How could you set up Live Rendering?**

The attribute @IBDesignable lets Interface Builder perform live updates on a particular view. IBDesignable requires Init frame to be defined as well in UIView class.

**2- What is the difference between Synchronous & Asynchronous task?**

Synchronous: waits until the task have completed Asynchronous: completes a task in the background and can notify you when complete

**3- Explain Compilation Conditions**

Compilation Conditions to use if DEBUG … endif structure to include or disable given block of code ve separate targets.

**4- What is made up of NSError object?**

There are three parts of NSError object a domain, an error code, and a user info dictionary. The domain is a string that identifies what categories of errors this error is coming from.

**5- What is Enum or Enumerations?**

According to Apple’s Swift documentation:

Managing state, the bits of data that keep track of how the app is being used at the moment, is an important part of a developing your app. Because enumerations define a finite number of states, and can bundle associated values with each individual state, you can use them to model the state of your app and its internal processes.

Enum is a type that basically contains a group of related values in the same umbrella but case-less enum won’t allow us to create an instance.

Reference: <https://developer.apple.com/documentation/swift/maintaining_state_in_your_apps>

**6- What is the bounding box?**

The bounding box is a term used in geometry; it refers to the smallest measure (area or volume) within which a given set of points.

**7- Why don’t we use strong for enum property in Objective-C?**

Because enums aren’t objects, so we don’t specify strong or weak here.

**8- What is @synthesize in Objective-C?**

synthesize generates getter and setter methods for your property.

**9- What is @dynamic in Objective-C?**

We use dynamic for subclasses of NSManagedObject. @dynamic tells the compiler that getter and setters are implemented somewhere else.

**10- Why do we use synchronized?**

synchronized guarantees that only one thread can be executing that code in the block at any given time.

**11- What is the difference strong, weaks, read only and copy?**

strong, weak, assign property attributes define how memory for that property will be managed.

Strong means that the reference count will be increased and

the reference to it will be maintained through the life of the object

Weak ( non-strong reference ), means that we are pointing to an object but not increasing its reference count. It’s often used when creating a parent child relationship. The parent has a strong reference to the child but the child only has a weak reference to the parent.

• Every time used on var

• Every time used on an optional type

• Automatically changes itself to nil

Read-only, we can set the property initially but then it can’t be changed.

Copy means that we’re copying the value of the object when it’s created. Also prevents its value from changing.

for more details check this out

**12- What is Dynamic Dispatch?**

Dynamic Dispatch is the process of selecting which implementation

of a polymorphic operation that’s a method or a function to call at run time. This means, that when we wanna invoke our methods like object method. but Swift does not default to dynamic dispatch

**13- What’s Code Coverage?**

Code coverage is a metric that helps us to measure the value of our unit tests.

**14- What’s Completion Handler?**

Completion handlers are super convenient when our app is making an API call, and we need to do something when that task is done, like updating the UI to show the data from the API call. We’ll see completion handlers in Apple’s APIs like dataTaskWithRequest and they can be pretty handy in your own code.

The completion handler takes a chunk of code with 3 arguments:(NSData?, NSURLResponse?, NSError?) that returns nothing: Void. It’s a closure.

The completion handlers have to marked @escaping since they are executed some point after the enclosing function has been executed.

**15- How to Prioritize Usability in Design ?**

Broke down its design process to prioritize usability in 4 steps:

• Think like the user, then design the UX.

• Remember that users are people, not demographics.

• When promoting an app, consider all the situations in which it could be useful.

• Keep working on the utility of the app even after launch.

**16- What’s the difference between the frame and the bounds?**

The bounds of a UIView is the rectangle, expressed as a location (x,y) and size (width, height) relative to its own coordinate system (0,0).

The frame of a UIView is the rectangle, expressed as a location (x,y) and size (width, height) relative to the superview it is contained within.

**17- What is Responder Chain ?**

A ResponderChain is a hierarchy of objects that have the opportunity to respond to events received.

**18- What is Regular expressions?**

Regular expressions are special string patterns that describe how to search through a string.

**19- What is Operator Overloading?**

Operator overloading allows us to change how existing operators behave with types that both already exist. Operators are those little symbols like +, \*, and /

**20- What is TVMLKit?**

TVMLKit is the glue between TVML, JavaScript, and your native tvOS application.

**21- What is Platform limitations of tvOS?**

First, tvOS provides no browser support of any kind, nor is there any WebKit or other web-based rendering engine you can program against. This means your app can’t link out to a web browser for anything, including web links, OAuth, or social media sites.

Second, tvOS apps cannot explicitly use local storage. At product launch, the devices ship with either 32 GB or 64 GB of hard drive space, but apps are not permitted to write directly to the onboard storage.

tvOS app bundle cannot exceed 4 GB.

**22- What is Functions?**

Functions let us group a series of statements together to perform some task. Once a function is created, it can be reused over and over in your code. If you find yourself repeating statements in your code, then a function may be the answer to avoid that repetition.

Pro Tip, Good functions accept input and return output. Bad functions set global variables and rely on other functions to work.

**23- What is ABI?**

ABIs are important when it comes to applications that use external libraries. If a program is built to use a particular library and that library is later updated, you don’t want to have to re-compile that application (and from the end user's standpoint, you may not have the source). If the updated library uses the same ABI, then your program will not need to change. ABI stability will come with Swift 5.0

**24- Why is design pattern very important ?**

Design patterns are reusable solutions to common problems in software design. They’re templates designed to help you write code that’s easy to understand and reuse. Most common Cocoa design patterns:

• Creational: Singleton.

• Structural: Decorator, Adapter, Facade.

• Behavioral: Observer, and, Memento

**25- What is Singleton Pattern ?**

The Singleton design pattern ensures that only one instance exists for a given class and that there’s a global access point to that instance. It usually uses lazy loading to create the single instance when it’s needed the first time.

**26- What is Facade Design Pattern?**

The Facade design pattern provides a single interface to a complex subsystem. Instead of exposing the user to a set of classes and their APIs, you only expose one simple unified API.

**27- What is Decorator Design Pattern?**

The Decorator pattern dynamically adds behaviors and responsibilities to an object without modifying its code. It’s an alternative to subclassing where you modify a class’s behavior by wrapping it with another object.

In Objective-C there are two very common implementations of this pattern: Category and Delegation. In Swift there are also two very common implementations of this pattern: Extensions and Delegation.

**28- What is Adapter Pattern?**

An Adapter allows classes with incompatible interfaces to work together. It wraps itself around an object and exposes a standard interface to interact with that object.

**29- What is Observer Pattern?**

In the Observer pattern, one object notifies other objects of any state changes.

Cocoa implements the observer pattern in two ways: Notifications and Key-Value Observing (KVO).

**30- What is Memento Pattern?**

In Memento Pattern saves your stuff somewhere. Later on, this externalized state can be restored without violating encapsulation; that is, private data remains private. One of Apple’s specialized implementations of the Memento pattern is Archiving other hand iOS uses the Memento pattern as part of State Restoration.

**31- Explain MVC**

• Models — responsible for the domain data or a data access layer which manipulates the data, think of ‘Person’ or ‘PersonDataProvider’ classes.

• Views — responsible for the presentation layer (GUI), for iOS environment think of everything starting with ‘UI’ prefix.

• Controller/Presenter/ViewModel — the glue or the mediator between the Model and the View, in general responsible for altering the Model by reacting to the user’s actions performed on the View and updating the View with changes from the Model.

**32- Explain MVVM**

UIKit independent representation of your View and its state. The View Model invokes changes in the Model and updates itself with the updated Model, and since we have a binding between the View and the View Model, the first is updated accordingly.

Your view model will actually take in your model, and it can format the information that’s going to be displayed on your view.

There is a more known framework called RxSwift. It contains RxCocoa, which are reactive extensions for Cocoa and CocoaTouch.

**33- How many different annotations available in Objective-C ?**

• \_Null\_unspecified, which bridges to a Swift implicitly unwrapped optional. This is the default.

• \_Nonnull, the value won’t be nil it bridges to a regular reference.

• \_Nullable the value can be nil, it bridges to an optional.

• \_Null\_resettable the value can never be nil, when read but you can set it to know to reset it. This is only apply property.

**34- What is JSON/PLIST limits ?**

• We create your objects and then serialized them to disk..

• It’s great and very limited use cases.

• We can’t obviously use complex queries to filter your results.

• It’s very slow.

• Each time we need something, we need to either serialize or deserialize it.

• it’s not thread-safe.

**35- What is SQLite limits ?**

• We need to define the relations between the tables. Define the schema of all the tables.

• We have to manually write queries to fetch data.

• We need to query results and then map those to models.

• Queries are very fast.

**36- What is Realm benefits ?**

• An open-source database framework.

• Implemented from scratch.

• Zero copy object store.

• Fast.

**37- How many are there APIs for battery-efficient location tracking ?**

There are 3 apis.

• Significant location changes — the location is delivered approximately every 500 metres (usually up to 1 km)

• Region monitoring — track enter/exit events from circular regions with a radius equal to 100m or more. Region monitoring is the most precise API after GPS.

• Visit events — monitor place Visit events which are enters/exits from a place (home/office).

**38- What is the Swift main advantage ?**

To mention some of the main advantages of Swift:

• Optional Types, which make applications crash-resistant

• Built-in error handling

• Closures

• Much faster compared to other languages

• Type-safe language

• Supports pattern matching

**39- Explain generics in Swift ?**

Generics create code that does not get specific about underlying data types. Don’t catch this article. Generics allow us to know what type it is going to contain. Generics also provides optimization for our code.

**40- Explain lazy in Swift ?**

An initial value of the lazy stored properties is calculated only when the property is called for the first time. There are situations when the lazy properties come very handy to developers.

Lazy properties are useful when the initial value for a property is dependent on outside factors whose values are not known until after an instance’s initialization is complete.

Lazy properties are also useful when the initial value for a property requires complex or computationally expensive setup that should not be performed unless or unitl it is needed.

**41- Explain what is defer ?**

defer keyword which provides a block of code that will be executed in the case when execution is leaving the current scope.

**42- How to pass a variable as a reference ?**

We need to mention that there are two types of variables: reference and value types. The difference between these two types is that by passing value type, the variable will create a copy of its data, and the reference type variable will just point to the original data in the memory.

**43- How to pass data between view controllers?**

There are 3 ways to pass data between view controllers.

1. Segue, in prepareForSegue method (Forward)

2. Delegate (Backward)

3. Setting variable directly (Forward)

**44- What is Concurrency ?**

Concurrency is dividing up the execution paths of your program so that they are possibly running at the same time. The common terminology: process, thread, multithreading, and others. Terminology;

• Process, An instance of an executing app

• Thread, Path of execution for code

• Multithreading, Multiple threads or multiple paths of execution running at the same time.

• Concurrency, Execute multiple tasks at the same time in a scalable manner.

• Queues, Queues are lightweight data structures that manage objects in the order of First-in, First-out (FIFO).

• Synchronous vs Asynchronous tasks

**45- Grand Central Dispatch (GCD)**

GCD is a library that provides a low-level and object-based API to run tasks concurrently while managing threads behind the scenes. Terminology;

• Dispatch Queues, A dispatch queue is responsible for executing a task in the first-in, first-out order.

• Serial Dispatch Queue A serial dispatch queue runs tasks one at a time.

• Concurrent Dispatch Queue A concurrent dispatch queue runs as many tasks as it can without waiting for the started tasks to finish.

• Main Dispatch Queue A globally available serial queue that executes tasks on the application’s main thread.

**46- Readers-Writers**

Multiple threads reading at the same time while there should be only one thread writing. The solution to the problem is a readers-writers lock which allows concurrent read-only access and an exclusive write access. Terminology;

• Race Condition A race condition occurs when two or more threads can access shared data and they try to change it at the same time.

• Deadlock A deadlock occurs when two or sometimes more tasks wait for the other to finish, and neither ever does.

• Readers-Writers problem Multiple threads reading at the same time while there should be only one thread writing.

• Readers-writer lock Such a lock allows concurrent read-only access to the shared resource while write operations require exclusive access.

• Dispatch Barrier Block Dispatch barrier blocks create a serial-style bottleneck when working with concurrent queues.

**47- NSOperation — NSOperationQueue — NSBlockOperation**

NSOperation adds a little extra overhead compared to GCD, but we can add dependency among various operations and re-use, cancel or suspend them.

NSOperationQueue, It allows a pool of threads to be created and used to execute NSOperations in parallel. Operation queues aren’t part of GCD.

NSBlockOperation allows you to create an NSOperation from one or more closures. NSBlockOperations can have multiple blocks, that run concurrently.

**48- KVC — KVO**

KVC adds stands for Key-Value Coding. It’s a mechanism by which an object’s properties can be accessed using string’s at runtime rather than having to statically know the property names at development time.

KVO stands for Key-Value Observing and allows a controller or class to observe changes to a property value. In KVO, an object can ask to be notified of any changes to a specific property, whenever that property changes value, the observer is automatically notified.

**49- Please explain Swift’s pattern matching techniques**

• Tuple patterns are used to match values of corresponding tuple types.

• Type-casting patterns allow you to cast or match types.

• Wildcard patterns match and ignore any kind and type of value.

• Optional patterns are used to match optional values.

• Enumeration case patterns match cases of existing enumeration types.

• Expression patterns allow you to compare a given value against a given expression.

**50- Explain Guard statement**

There are three big benefits to guard statement.

One is avoiding the pyramid of doom, as others have mentioned — lots of annoying if let statements nested inside each other moving further and further to the right. The second benefit is providing an early exit out of the function using break or using return.

**51- Explain Tuple in swift**

Tuple is a group of differentvalues represented as one. Accounting to apple, a tuple type is a comma-separated list of zero or more types, enclosed in parentheses. It’s a miniature version of a struct.

**52- Explain Memory Leak in Swift**

**Prevent Retain cycle**

**53- ViewController Lifecycle**

LoadView()->ViewDIdLoad()->ViewWillAppear()->ViewDidAppear()-> ViewWillDisAppear()->ViewDIdDisAppear()

**54- Memory Leak**

A memory leak occurs when a given memory space cannot be recovered by the system because it is unable to tell if this memory space is actually in use or not.

**55- Memory Management**

Swift uses the ARC(Automatic Reference Counting) memory management model

An object is retained in memory by incrementing its reference count, and then released by decrementing that same count. Once an object’s retain count reaches zero means that object is deallocated from memory.

56- escaping closure

A closure is said to escape a function when the closure is passed as an argument to the function, but it called after the function returns.

57- multithreading

Grand Central Dispatch & NSOperation Queues

Serial queue: Task complete in order(slower) Main Queue

Concurrent queue: faster but unpredictable order Background Queue

Grand central dispatch:

DIspatchQueue.main.async{

Self.tableView.reloadData()

}//download json data from internet(background thread) shift to main thread and update UI

DispatchQueue.global(qos: .background).async{

//code to run on background queue

}//Move stuff to background queue manually