(01)

1) **Describe what a Native app is and what a Hybrid app is.**

Native apps for specific OS and hybrid apps can run on both android and iOS.
Hyper apps implemented using HTML, CSS, javascript and Native apps implemented using a specific technology.

2) **Explain the main advantages and disadvantages ofNative and Hybrid apps.**

Native Apps	Hybrid Apps
Offer speed	Enhanced UX
Work Offline	Sametime features release for both OS
Expensive	Cheaper Development Cost
No flexibility	Higher flexibility

3) **What are the main disadvantages of developing apps for iOS?**

Expensive development

Time consuming development
Lengthy downloading process
Applications maintenance is expensive
Requires frequent updates

4) **What are the reasons for introducing Swift for developing iOS apps and what are the main disadvantages of using Swift?**

Swift Advantages	Swift Disadvantages

Easy code maintenance	Language is quite long
Easy code readability	Limited talent pool
Speed app development	Lack of support for earlier iOS
Supported by dynamic development	Poor interact with third party tools
Supports dynamic dispatch	Language is still new
Supports late binding	If you don't have any prior knowledge on swift it is difficult to follow.

5) **List down the main iOS Development Prerequisites.**

Xcode
Mac computer

Intel based or M1 processor running on a compatible mac os version
Knowledge about Swift

6) Briefly explain the Apple recommended architecture pattern for developing apps?

MVC,- Model view controller

Models store your data, such as the names of products in a store.
Views render data for users, for example a table showing the list of products available.
Controllers combine the two, by querying the model and converting its data to something views can show.

MVVM, - Modal View ViewModal

In here presentation and business parts communicate with each other using reactive binding called view and view modal.

VIPER - **V**iew Interactor **P**resenter **E**ntity **R**outing

It is the Cleanest Architecture to iOS apps. The word VIPER stands for View, Interactor, Presenter, Entity, and Routing.
Clean Architecture divides an app's logical structure into layers of responsibility.

(02)

1) What are the main functionalities of Xcode?

Source Editor
Integrated Build System
Graphical debugger
Complete documents
Assistant editor

2) List down the main features and main frameworks of Cocoa Touch.

Main features	Main Framework
Storyboards	Foundation kit
Auto layouts	StoreKit
Multitasking	MapKit
App extension	GameKit
Airdrop	UIKit

3) What are Lazy Properties in Swift?

A lazy stored property is a property whose initial value isn't calculated until the first time it's used.

```
struct Person {
   let name: String
   let age: Int
}

struct PeopleViewModel {
```

```
    let people: [Person]

    lazy var oldest: Person? = {
        print("Lazy var oldest initialized")
        return people.max(by: { $0.age < $1.age })
    }()

    init(people: [Person]) {
        self.people = people
        print("View model initialized")
    }
}
```

```
class DataImporter {

    var filename = "data.txt"

}

class DataManager {
    lazy var importer = DataImporter()
    var data = [String]()

}

let manager = DataManager()
manager.data.append("Some data")
manager.data.append("Some more data")
```

4) What are Type Properties in Swift?

Class
Struct
Enum

Use static keyword

5) **Provide example code for usage of Optional Binding and Optional Guard.**

```
func greet(person: [String: String]) {

guard let name = person["name"] else {
return
}

print("Hello \(name)!")
guard let location = person["location"] else {
print("I hope the weather is nice near you.")
Return

}

print("I hope the weather is nice in \(location).")

}

greet(person: ["name": "John"])

greet(person: ["name": "Jane", "location": "Cupertino"])
```

03)

01) **Briefly explain why Generics are used in Swift.**

Generics code enables us to write flexible and reusable functions and any type of requirements that we define.

02) **Write the code for a Generic data structure that has two parameters, namely 'key' and 'value'.**

```
struct something<A, B> {
        let key:A
        let value:B

          add(b: B);
          delete(a: A){

          };
```

}

Var cheeseBall = something<String, Int>
cheeseBall.key = “Name”

cheeseBall.value = 5
cheeseBall.delete(“Name”)

Var flutter = something<Int, String>

Flutter.key = 1
Flutter.value = “Cat”

flutter.delete(1)

03) **What is a mutable array? What happens in the mutable array when we remove or add a value to a mutable array with a valid index?**

If we create an array to a variable it is a mutable array, which means we can add, edit, and remove values.
But if we create an array to a constant, it is immutable.
If we add a value it will be added.

```
students.insert("Liam", at: 3)
students.removeLast()
```

03) What happens if we try to initialize a Set in the shorthand format instead of writing the full initialization code?

```
var mySet = Set<Int>()
```

we can create an empty set of a certain type. We can write the code in fewer lines.

04) What is the name of the Garbage Collection process found in iOS? Briefly explain how it works.

ARC

There’s no garbage collection methods. Now Automatic Reference Counting (ARC) is used to compile in time and reference counting. **ARC** is a compile time feature XCode provides us for automated memory management.

05) **Explain what is a retain cycle and the mechanism used to avoid retain cycles.**

A memory leak in iOS is when an amount of allocated space in memory cannot be deallocated due to retain cycles. Since Swift uses Automatic Reference Counting (ARC), a retain cycle occurs when two or more objects hold strong references to each other. As a result these objects retain each other in memory because their retain count would never decrement to 0, which would prevent deinit and memory from being freed.

To avoid retain cycles we Use "close" methods to break **retain cycles**.

Use weak pointer to parent , which means a child should be using weak reference to parent, which is not **retained**.

04)

01) What are the three main design themes provided by Apple? Briefly explain each.

02) What are the three main navigation styles recommended by Apple's HIG?

Hierarchical Navigation - Make one choice per screen until you reach a destination.

Flat Navigation - Switch between multiple content categories. Music and App Store use this navigation style.

Content-Driven Navigation - Move freely through content.

03) What is the difference between Tables and Collections when developing user interfaces? Mention best usages for each.

Collection views provide the same general function as table views except that a collection view is able to support **more than just single-column layouts**.

A table view displays a list of items in **a single column.**

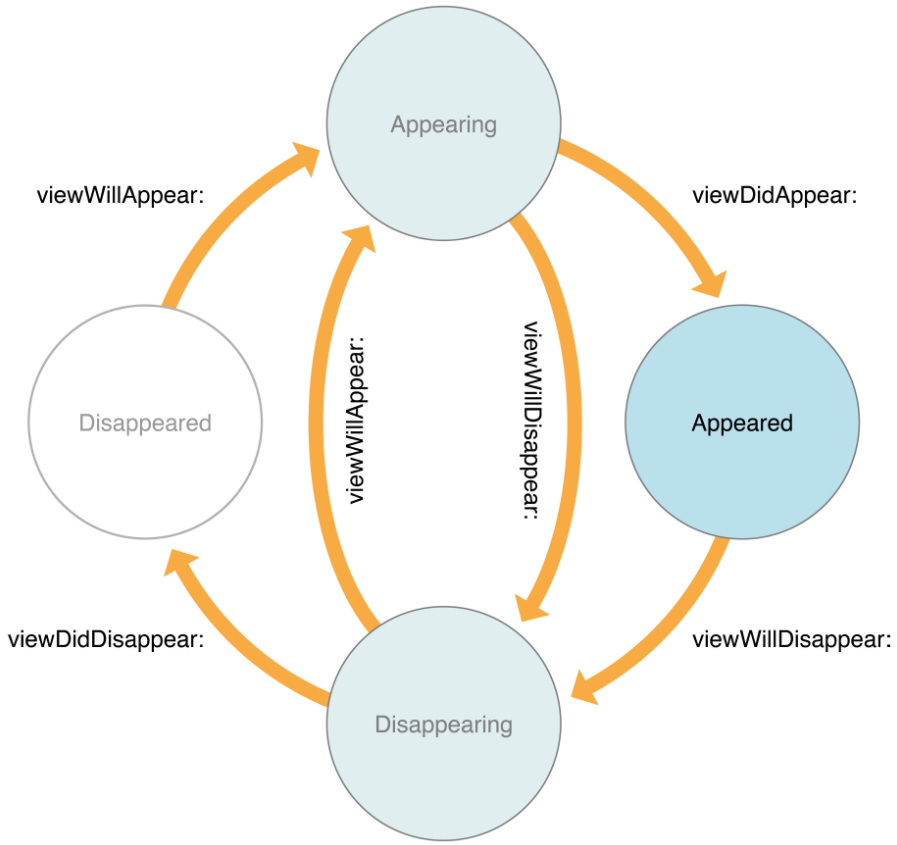
04) Explain Cropping, Letterboxing and Pillarboxing.

Cropping - The cropping controls allow you a great degree of flexibility and versatility.

Letterboxing -

Pillarboxing -

05) Briefly explain the life cycle of a ViewController.



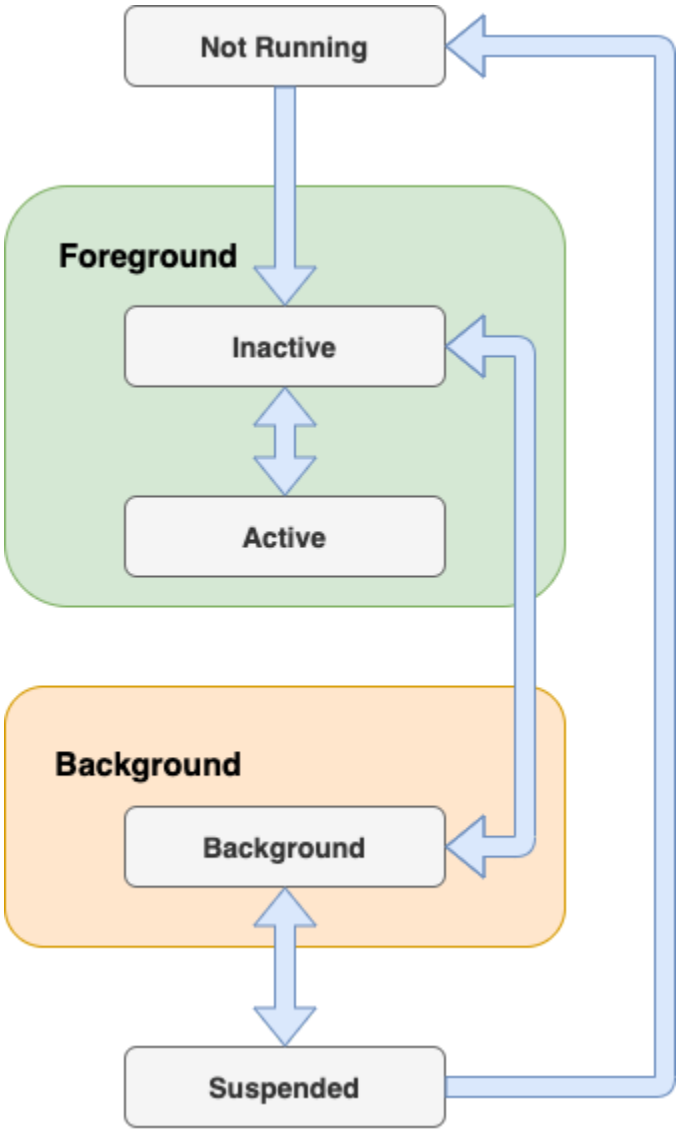
viewWillAppear - This Method is called every time before the view are visible to and before any animation are configured.

viewDidAppear - This Method is called after the view present on the screen.

viewWillDisappear: - This method called before the view are remove from the view hierarchy.

viewDidDisappear - Use this method to stop listening for notifications or device sensors.

06) Briefly explain the life cycle of an iOS application.



1. Not Running: the app is considered to be in a Not Running state when it is terminated by the user.
2. Inactive: the app is in an inactive state when it is in the foreground but receiving events.
3. Active: it is a normal mode for the app when it is in the foreground state and receiving all the user events.
4. Background: the app transitions into the background state when the user taps on the home screen while using the application.
5. Suspended: in this state, the app remains in the background and doesn't execute the code.

05)

01)What are the main usages provided by the Instruments tool present in XCode?

Xcode Instruments is best described as a powerful and flexible performance-analysis and testing tool.

We can launch instruments.
Create , save, and open trade documents
We can access and use instruments
Configure instruments display settings

02) Name access control types found in Swift and briefly describe each.

Public - access from anywhere
Private - only access within file
Internal - only access within source module

03)Name four Set Operations and four Set Membership Methods found in Swift.

Set Operations- Subtract, Union, Intersection, isDisjoint

Set Membership - contains, isSubset, isSuperset, isStrictSubset, isDisjoint

04) What is meant by Protocol Inheritance?

One protocol can inherit from another in a process known as protocol inheritance.
We can inherit from multiple protocols at the same time before you add your own customizations on top.

05)What are Closures in Swift?

Closures in Swift are similar to blocks in C and Objective-C can be passed around and used in your code
C.losures can capture and store references to any constants