1. **enum** in swift is used to defines a common type for a group of related values.
2. **states** of the common execution can be as follows:

* Not running – This state means that there is no code that is being executed and the application is completely switched off.
* Inactive – This state means that the application is running in the background and is not receiving any events.
* Active – This state means that the applications are running in the background and are receiving the events.
* Background – This state means that the application is executing the code in the background.
* Suspended – This state means that the application is in the background and is not executing.

1. **Object lifecycle**
2. **AppDelegate Lifecycle**
3. **UIViewController Lifecycle**
4. **@synthesize** – It generates the getter and setter methods for the property.
5. **@dynamic** – It notifies the compiler that the getter and setter are implemented at some other place
6. **deinitializer** is used to free up the resources by deallocates your instance class when they remain no longer useful for you. a **dinit keyword** is used to represent it. It is only present on class types.
7. **Nil & Null** - there is no difference between nil and None in Swift. nil & .none are the equivalent to each other.
8. **Generic** is efficient to write sensitive and reusable functions and types.
9. **How to call Swift Code from Objective - C.**
10. **How to call Objective - C code from Swift.**
11. **Switch & If statement - which is better, why?**
12. **Trailing - Closure? & Block?**
13. **Protocol? Can I create an instance of protocol?**
14. **Double ?? mark? When do I need to use it?**
15. **Array - How to remove or get a unique value from an array?**
16. **Optional Chaining & Optional Binding Difference?**
    1. Changing (Safe Use) **-** something?.somevalue?.somemethod()
    2. Binding -

Var name: String? = “Swift”

If let newName = name {

print(newName)

}

1. **CoreData-NSManagedObject ->** The core unit of coredata.These are model objects that hold data. Set the attributes and relationships.
2. **NSManagedObjectContext ->** CRUD Operation
3. **NSManagedObjectModel ->** describes the application entities
4. **NSPersistentContainer ->** Data load from entity
5. **NSPersistentStoreCoordinator ->** Reading & writing
6. **MVC?**
   1. **Model**: A model represents the app’s data. It stores info, such as products in a store. A model manages the state of the application.
   2. **View**: A view is responsible for showing and interacting with the UI. For example, a view renders a table of products for your app’s user.
   3. **Controller**: The controller is what glues the model and the view. It’s responsible for controlling the logic that goes between the two.
7. **Inout -** change function parameters (let -> convert -> var)

func change(\_ number: **inout** Int) {

number = 2

}

var num = 1

change(&num) //how to pass value

print(num) // 2

23.== & === difference

* 1. **==** is the equality operator.
  2. === is the identity operator.

24.**Payment Gateway -**

**(**Strip, Paytm, ApplePay, In-app (subscription))

25.**In-app Purchase**

1. Delegate - SKProductRequestDelegate,
2. Observe - SKPaymentTransactionObserve
3. Product Request
4. PaymentQueueRestoreCompletedTransactionFinished
5. PaymentQueue -> purchased, failed, restored

**26**. **PushNotification - Willset & Didset**

**27. .Pem file? Where & Why use?**

**28. Singleton Class in objective -c.**

**29. Pass data**

**30. Socket**

**31. Target -> Extension Certificate?**