Weak self ... weak variable
Weak variable and weak self that is used to where you want to avoid creating a strong reference cycle. the weak variable will automatically set to nil. And that is a class protocol bounded

Strong variable in swift
a strong variable is the default reference type, and strong reference keeps a strong reference on an object. That is a default inbuilt data in struct

Unowned Variable in swift
an unowned variable is a type of weak reference that does not hold a strong reference to the instance, an unowned reference assumes that the referenced instance will always be available

//
Classes: Classes are reference types that allow you to define complex data structures. They support inheritance, enabling you to create class hierarchies. Objects created from classes are passed by reference, meaning that multiple variables can refer to the same object instance.
Structures: Structures, also known as structs, are value types. They are used to create more lightweight objects and are typically copied when assigned or passed around. Unlike classes, structures do not support inheritance.
Enumerations: Enumerations, or enums, are value types used to define a group of related values. Each case of an enumeration can have associated values. Enums can also have methods and computed properties.
Closures are self-contained blocks of functionality that can be passed around and used in your code. And this is a reference type

App Life cycle
The iOS app lifecycle refers to the sequence of events and states...
Not Running: The app is not running or has been terminated by the user or the system.
Launching: The app is being launched,
In Foreground:
Inactive: The app is active but not receiving events,
Active: The app is in active, and receiving the events

4 Background:
Background: The app enters the background when the user switches to another app
Suspended: The app is still in the background but is no longer executing code.
5 Terminating: The app is about to be terminated by the user or the system.
Methods UIApplicationDelegate...
application(_:didFinishLaunchingWithOptions:): Called when the app finishes launching.

applicationWillResignActive(_:): Called when the app is about to become inactive.
applicationDidEnterBackground(_:): Called when the app enters the background.
applicationWillEnterForeground(_:): Called when the app is about to enter the foreground.
applicationDidBecomeActive(_:): Called when the app becomes active.
applicationWillTerminate(_:): Called when the app is about to terminate.

Mutating keyword.
In Swift, the mutating keyword is used to indicate that a method or function can modify the properties of a value type, such as a structure or an enumeration.

Defer statement
In Swift, the defer statement is used to define a block of code that is executed just before the current scope is exited

CoreData
Core Data is a framework provided by Apple that allows developers to manage the model in application. It provides an object-oriented approach to interact with a persistent store, such as a SQLite database, like save permanent data in locally use Application.
1> Core Data Model: that is used to create entities and attributes and relationships etc
2> Managed Object Context: that is used to interact persistent store for create and updating and deleting data in core data managed object.
3> You can use NSEntityDescription to create instances of entities defined in your Core Data model and new insert entities.
4> NSFetchRequest that is used to fetch object , as well as any predicates, and o filter and sort the results.
What is the difference between a managed object context and a managed object model?
Moc that is used to creating, updating deleting data in sqlite persistence store.
Mom is a overall structure of core data model or schema
// core data delete rules->
1> Cascade: when you delete user table and then automatically delete all relation ship user table is called cascade delete
2> No Action: when you delete user table that user table has been deleted but other relationship but don’t effect relations ship tables
3> Nullify: when you deleted user table then all relation ship set to parent nil
4> Deny: when you delete user table but not deleted then propagate error firstly you will delete all relationship child of user table after that you will delete user.
//
CoreData Relationship->
Four type of relationship of core data

1> One To many -> for you have user table but user table attach multiple address that one user many address
2> many to one-> for you have address table but address attach to the user table that multiple address attached one user is called many to one
3> one to one-> for you have user table that attach passport table that is one to one because one user have one passport and one passport have one user.
4> many to many-> for you have student table that every student attach course table is called many to many relationship.
//
Properties in core data transient and optional
Transient property means that is not persists in database when you save data model. It is like compute because that property calculate value by drive value but don't save database for example age based on dob.
5> persistence store coredinator —The persistent store coordinator is responsible for managing the persistent store,
6> ConcurrencyType: NSMainQueueConcurrencyType... It is used to create a managed object context that operates on the main queue
7> NSPrivateQueueConcurrencyType : It is used to create a private queue-based managed object context. Private queue are suitable for performing background tasks or operations
8> Fetch requests are primarily used to fetch objects from the persistent store based on

Push Notifications iOS

To implement push notifications in an iOS app, you need to follow more steps:
1> Enable push notifications in your App ID: In the Apple Developer Portal
2> we will create apns Authentication key and apple push Notifications certificates that is used established secure connection between your app and apns server
3> enable push notifications capabilities in your project
4> Create a new Firebase project or use an existing and create app in use your app bundle id and download plist file integrate in your project
5> after that we will finished app setting like we will attach apple developer account team id and Authentication file and keyid and

And after that APNS Working
App will send fcm token to server
And server will send notifications with fcm token to apns server
And then apns server will send notification to app

There are following methods use in push Notifications
didReceiveRemoteNotification—
method to handle incoming remote notifications.
didReceiveRegistrationToken:
That is used to get fcm token and device token
willPresent notification:

That is used to get push notifications data when your app is in the foreground
didReceive notification: that is used to get data when notification is tapped
handleNotificationPayloadOnLaunch:
This method is used to handle notification payload if user launch app from backgroud

// MVC is a design pattern that consists three components model- view- controller
// the model is basically data
// the view is simply user interface ui whatever you the see on the screen
// controller that is used to connect each other view and model

generics defination = that is used reusablity function that can be used any type in swift
Core Location framework provide by apple that is used to get location data. Such as device current location, change location and Reverse Geocoding
1> get device's current location you will need to create instance allocation manager and check permissions requestwheninuseauthorization
And there allocation delegates method
Didupdatelocation: that is used to handle updated last long
Didfailwitherror: that is used to handle error
And after that provide kay infoplist NSLocationWhenInUseUsageDescription that required for check location permissions
2>background mode by capabilities
...what is background fetch in background mode
background fetch that is used to allows your app to periodically fetch new data and update content in the background
3> what is background processing in background mode
Background processing that is used to allows your app to continue executing tasks in the background,
4> what is location update in background mode
Location update that is used to allows an app to continue receiving location data even when the app is not in the foreground
5> remote notifications that is used to allows an app to receive notifications from a remote server, even when the app is not in the foreground

//What do you mean by property in iOS?
//What are different types of iOS Application States?

///Deep linking in ios that used to directing users to move specific location within a mobile app by using custom URL Schemes:, Universal Links:

automatic reference counting (ARC) is used to manage memory mgmt it is work allocate and deallocate object from memory and strong reference will be automatically handle arc and weak and unowned reference will be handle by developer.

Cocoa Cocoa Touch
It is an application framework for building applications that run on Mac OS X.
It is the application framework for building applications that run on devices like iPhones and iPad.
Cocoa frameworks includes such as Foundation and AppKit Cocoa Touch frameworks includesFoundation and UIKit

Enumeration is a user-defined data type that allows you to define a group of related values. Each value within an enumeration is called a case

he Memento pattern is a behavioral design pattern that allows you to capture and restore the internal state of an object without violating encapsulation.

Memento pattern is a behavioral design pattern that allows you to capture and restore the internal state of an object without violating encapsulation. It is useful in scenarios where you need to save and restore the state of an object, such as undo/redo functionality

iOS supports SBJson framework.
SBJson is a JSON parser and generator for Objective-C.
It provides flexible APIs and additional control, making JSON handling easier

SBJson is a data interchange formatter that's easy to read and write. It's one kind of JSON. This framework is supported by IOS. Where as JSON means JavaScript Object Notation.It is syntax for storing and exchanging information or data

Which is the application thread from where UIKit classes should be used?
UIKit classes should be used only from an application’s main thread.

When would you say that an app is not in a running state?
An app is said to be in ‘not running’ state in the following cases:
– When it is not launched.

– When it gets terminated by the system during running.

Outline the class hierarchy for a UIButton until NSObject.
UIButton inherits from UIControl, UIControl inherits from UIView, UIView inherits from UIResponder, UIResponder inherits from the root class NSObject.

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

To dismiss multiple view controllers and navigate back to a specific view controller in a Swift app, you can make use of the popToViewController(_:animated:) method provided by the navigation controller.

The main components of an iOS app include:
App Delegate: The AppDelegate class is the entry point and handles crucial app-level events and lifecycle callbacks
View Controllers: View controllers are responsible for managing the user interface and interaction logic.
Views: Views are the visual elements that make up the user interface of an app. They can be buttons, labels, text fields, images, or custom UI components
Storyboards or Interface Builder: Storyboards or Interface Builder is a visual tool in Xcode that allows developers to design app interfaces using a drag-and-drop approach.
Model: The model represents the underlying data and business logic of the app. It encapsulates data structures, algorithms
Views and Controls: These are UI elements that the user interacts with, such as buttons, labels, text fields, sliders, and switches.
Networking: iOS apps often communicate with web services or APIs to fetch data or perform actions. Network components,
Persistence store: That is used to save data locally in your iOS app like sqlite database.

What are the different types of storage options available in iOS?
UserDefaults: UserDefaults provides a simple way to store small amounts of data, such as user preferences, settings
Keychain: The Keychain Services API allows you to securely store sensitive data such as passwords, authentication tokens, certificates
Core Data
iCloud: iCloud is Apple's cloud storage and synchronization service that enables data synchronization across multiple devices.

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.

Singleton is a design pattern that only one instance of the class exists throughout the application

JSONSerialization class provides functionality for converting between JSON data and Foundation objects

What is Observer Pattern?
In the Observer pattern, one object notifies other objects of any state changes

MVVM (Model-View-ViewModel) is a popular design pattern used in iOS development. It provides a structured approach to separating the user interface

//MVVM is a design pattern that consists three components model-viewModel-view
// view is a whatever display on the screen,
// model is consist basically data
// viewModel that is used to interact model and view and that is used to write business logics
// view can talk to the viewModel, and viewModel can talk to view but view cannot talk to model,
// viewModel can talk model, and model can talk to viewModel
Advantage of mvvm separation of concern and testability
Disadvantage improve enhanced separate api calls
And suppose you can change model then automatically update ui suppose you can change view then automatically update ui.

KVC and KVO are both concepts related to Cocoa and Cocoa Touch frameworks used in Apple's macOS and iOS development. They stand for Key-Value Coding (KVC) and Key-Value Observing (KVO),
Key-Value Coding (KVC):
KVC is a mechanism that allows you to access the properties of an object indirectly using keys or key paths. Instead of directly accessing properties
Key-Value Observing (KVO):
KVO is a mechanism that allows objects to observe changes in the properties of another object. With KVO, you can register an object to receive notifications when a specific property of another object changes

Swift is the fastest-growing programming language today, created by Apple. With a comparison objective c aSwift implements all the features of other

modern languages, you can find type inference, optional, generics, and such higher-order functions. It is compatible with macOS, iOS, watchOS, and tvOS.

Optional A type that represents either a wrapped value or the absence of a value.
to declare an optional, you append a question mark (?) When a value is absent, the optional is set to nil.

There are several ways to unwrap an optional

Optional binding-> You can use optional binding with if let or guard let statements
Forced Unwrapping: You can use the force unwrap with exclamation mark operator (!) to forcefully unwrap if value is exists that is correctly your code run or you are facing run time error

Lazy properties are properties whose initial value isn't computed until the first time they are used.
Including the lazy modifier/keyword before the declaration of a stored property indicates it is lazy

In Swift, reference type instances share a single copy of their data, so that every new instance will point to the same address in memory. A typical example is a class , function , or closure

Value types in Swift. The basic types in the Swift programming language are value types, and they include arrays, Booleans, dictionaries, floating-point numbers, integers, and strings.Every instance of a value type has unique data that users may access and modify directly. This data is its value

AVFoundation
Work with audiovisual assets, control device cameras, process audio, and configure system audio interactions.

Swift 4 also introduces Tuples type, which are used to group multiple values in a single compound Value. The values in a tuple can be of any type, and do not need to be of same type

What are the most important features of swift?
Answer: Some important features of swift are given below:
* More impressive structs and enums
* Protocol oriented
* Optional Types

* Type Safety
* Not required to use semicolons
* Safe by default
* Less code, fewer files
* Forced Unwrapping
* Tuples
* Closures
* Much faster when compared to other languages.

PLIST stands for Property List. PLIST is basically a dictionary of value and keys that can be stored in our file system with a .plist file extension. The property list is used as a portable and lightweight means to store a lesser amount of data. They are normally written in XML.

Define Access control ?
More public private file private

Final keyword
When you declare a class as final, it means that it cannot be inherited by another class

typealias in swift
In Swift, typealias is used to define an alternative name for an existing type. It allows you to create a custom name that can be used interchangeably with the original type.

Type safety & Type Inference ?
Swift is a type-safe language agar app kisi variable ka type define card ga to usko koi or type ki value doge to vo asign nahi hogi

Type Inference: If you don’t specify the type of value you need, Swift uses type inference to work out the appropriate type
var meaningOfLife = 42 // meaningOfLife is inferred to be of type Int
meaningOfLife = 55 // it Works, because 55 is an Int

Dependency injection is a design pattern that is used inject one view controller to other like init injection and function param injection etc

Dependency inversion that is used to inject one view controller another by using protocol that is a protocol inject

atomic means only one thread access the variable (static type)

nonatomic means multiple thread access the variable (dynamic type).

Access control
Open: that is used to access own module and another inherit module and another model sub classes access
Public: similar to open access modifier but we can't inherit another module subclasses
Internal: This is the default access level. Internal entities can be accessed from any source file within the module but not from outside the module.
File Private: Entities with file-private access can only be accessed from within the same source file
Private: The most restrictive access level, limiting the use of an entity to the enclosing declaration (class, structure, or extension)

Final keyword
final, it means that its implementation is complete and cannot be modified or extended by any subclass.

```
class BaseClass {
    final func myMethod() {
        // Method implementation
    }
}


class TestOpen: BaseClass{
// not access my method because this method marked final
}
```

Escaping closure: escaping closure is called after function return and escaping closure block used self is mandatory
non-escaping closure is called before function return and non escaping closure block used self is optional

Map method that is used to mapping and transforming data in swift
Flat map that is used to remove nil and optional in array and remove extra array inside array swift
filter to loop over a collection and return an Array containing only those elements that match an include condition.
Use reduce to combine all items in a collection to create a single new value.

Value type... that are many type struct enum string int double, value type ko

app copy Kar sakte ho kiss dosra variable me assign kar sakte ho

Reference types are not copied when assigned to a new variable or passed as a function argument. Instead, they create a reference to the existing instance in memory. Class and closure

Optional chainning
Optional chaining is a process for querying and calling properties, methods, and subscripts on an optional that might currently be nil. If the optional contains a value, then the property, method, or subscript call succeeds; and if the optional is nil, then it returns nil.

```
struct Person{
   var address: Address?
}

struct Address{
   var street: String
}

let person: Person? = Person(address: Address(street: "knk"))

let street = person?.address?.street
print(street)
```

Associated type me you can declare placeholder type inside protocol then you can adopt protocol then you will set type and automatic get type in swift Associatedtype Data

Adaptor design patter that is structural design pattern that allow work two incompitable interfaces to work together

Factory design Pattern is a creational design pattern that is used to create object without specifying exact class of object that will be created.

Protocol that is used to define blueprints methods properties a that can be adopt another class struct etc

And
Protocol extension it used to provide default implementation methods computed property of an existing protocol is called protocol extension

Get and Set in protocol->

when you declare a property in a protocol with both get and set, you are defining a property requirement that can be both read (get) and written (set) by types conforming to the protocol.

```
// Define a protocol named "Animal"
protocol Animal {
    var name: String { get set }
    func makeSound()
}

// Create a struct that conforms to the "Animal" protocol
struct Dog: Animal {
    var name: String

    func makeSound() {
        print("\(name) says Woof!")
    }
}

struct Cat: Animal {
    var name: String

    func makeSound() {
        print("\(name) says Meow!")
    }
}

// Usage
let myDog = Dog(name: "Buddy")
let myCat = Cat(name: "Whiskers")
```

myDog.makeSound() // Output: Buddy says Woof!
myCat.makeSound() // Output: Whiskers says Meow!
//

Uiviewcontroller life cycle in swift

Initialization
View did load that is called one time your view is loaded
View will appear that is called before the is bout added in to view hierarchy
View did appear that is called after view has been added into view hierchachy
View will disappear that is called before view has Been removed view hierchachy
View will didappear that is called after view has been removed to the view hierchachy
Deinit that is used to that view deallocated

Control transfer statements is swift
Break that is used to inside loop that is immediately exits to the loop
Continue that is used to inside loop because that the iteration will continue
Return that is used to return statement and value to the caller side
Fall through that is used to match cases continue

Auto closure that is used to wrap an expression and remove curly braces inside function params
Trailing closure that is used to write a closure as a function last argument is called trailing closure.

Declarative Programming that is used to what do you achieve rather then how to do like swiftui

//Functional Programming function take a argument as a function is called functional programming like map filter reduce sorted etc

Imperative Programming that is used to how to do achieve goals, by usnig step by step instructions execute. Like uikit

Reactive Programming that is used to transform data like suppose we will change one side then another side update automatically is called reactive programming like rxswift combine observable object in swiftui

PopToViewcontroller that is used to navigate specific controller and one previous controller we can get view controllers array and then navigate your choice view controller

```
//access specifiers
public class PublicClass {
    // This class is accessible from any module.
}

open class OpenClass {
    // This class is accessible from any module and can be subclassed and overridden.
}

internal struct InternalStruct {
    // This struct is accessible only within the module it is defined in.
}

fileprivate enum FilePrivateEnum {
    // This enum is accessible only within the same source file.
}

private func privateFunction() {
    // This function is accessible only within its own scope (e.g., the same class or file).
}
//

//
```

Queues answer

Queues are part of the Grand Central Dispatch (GCD) framework
Queues allow you to execute tasks in parallel to improve the performance of your app
That provide a multiple queues.
Main Queue: The main queue is a serial queue that runs on the main thread.

```
DispatchQueue.main.async {
    // Perform UI-related tasks here
}
```

Global Queues: The global queue that runs on the background thread.

```
let backgroundQueue =
```

DispatchQueue.global(qos: .background)
backgroundQueue.async {
    // Perform background tasks here
}
Serial Queues: A serial queue executes tasks in the order they are added to the queue. Each task starts only after the previous one finishes.
let serialQueue = DispatchQueue(label: "com.example.serialQueue")

serialQueue.async {
    // Task 1
}

serialQueue.async {
    // Task 2 (starts after Task 1 finishes)
}

Concurrent Queues: A concurrent queue allows multiple tasks to run simultaneously,
let concurrentQueue = DispatchQueue(label: "com.example.concurrentQueue",
attributes: .concurrent)

concurrentQueue.async {
    // Task 1 (can run concurrently with other tasks)
}

concurrentQueue.async {
    // Task 2 (can run concurrently with other tasks)
}

//
Auto Layout is a powerful layout system provided by Apple's UIKit framework for building user interfaces in iOS applications. It's designed to create adaptive and responsive user interfaces that can automatically adjust to various screen sizes, orientations, and

devices.
//
Can we de-init a singleton object, if yes then give a real-life example where we need to de-init the singleton object? in ios
//

//
How do you manage security for applications? and how you will apply security in API calls
Answer
Authentication and Authorization:
API Keys or Tokens
App Store Guidelines:
//
VIPER (View-Interactor-Presenter-Entity-Routing): design pattern example ios
VIPER is an architectural pattern that aims to separate concerns and create a clear division of responsibilities in an iOS application.

//what is Recursive function in swift
That function called itself like factorial is called recursive function jo function apne app ko call karta hai wo function recursive function kahlata hai

//
Upcasting in swift
Upcasting means you can assign a sub class instance to a super class instance

```
Class Animal{}
Class Dog: Animal{}
Let dog: Dog = Dog()
Let animal: Animal = dog//upcasting
```

//Downcasting

Downcasting means you can assign a super class instance to a sub class instance
To perform downcast as? As!

```
Class Animal{}
Class Dog: Animal{}
Let animal: Animal = Dog()
If let dog = animal as? Dog{
// down casting safe
}
```

//Logical operator in swift
Logical operators are used to check whether an expression is true or false . They are used in decision-making. For example, var a = 5, b = 6 print((a > 2) && (b >= 6)) // true.

//Loops in swift

For Loop-> a for loop is used to iterate over a sequence like array collection is called for loop.

```
For I in 1...5{
print(i)
}
```

While Loop> a while loop that is repeatedly executes a block of code as long as a specified condition is true.

```
 Var I = 0
While I < 5{
Print(i)
I += 1
}
```

Repeat while loop>
A repeat while loop is similar to do while loop. It execute a block at least once without check condition.

```
Var j = 0
```

Repeat {
Print(i)
J += 1
} while j < 5
//

//Init
Designated Initialisers every class have a designated initiliazers if you will write base insert from super class then you write a designated init then firstly called our property and set after that called super class properties
Convince. Init means you will write an another init with different implementation brother of designated init must called self.init firstly after that called own property.

Failable Init and Memberwise init

Member init means struct provide default init is called member init.

Failable init means suppose you will create an optional initializer is called bailable init.

Try means that is used called throws function that function catch error by using catch block
Try? That is used to error convert into nil
Try! That is used unwrap optional value that sure the will not get

Nil collision operation ?? That is used unwrap optional value and provide default value
? That is used declare optional and optional chaining calling property
! That is used declare option force unwrap and unwrap

optional value by force unwrapping

//
Synchronous and ayschronus
Means
Sysnchronous task means the second task will not execute until the first task will not complete is called sysnchrnous programming like non escaping closure
Asynchronous programming that task execute independently the code does not wait the first task is complete task execute are parallel like escaping closure
//

Grand Central Dispatch (GCD) is a low-level API provided by Apple for manage concurrent operations like asynchronous task, it allow you to perform task in background thread.

GCD Used Dispatch queues for manage the execution of task.
Two type of queue.
Main Queue-> that is used to execute task in main thread
Global Queue-> that is used to execute task in background thread.

Now let's add the @IBDesinable signature to the CustomView class, implement the prepareForInterfaceBuilder() method, and update the init(frame:) initializer:

3- Explain Compilation Conditions
Compilation Conditions to use *if DEBUG ...*
endif **structure to include or disable given block of code ve separate targets.**

What is the difference between atomic and nonatomic synthesized properties?
Ans. Atomic:
Default behaviour

Not fast
Make certain the process is completed entirely by the CPU

Nonatomic:
Not default behaviour
Faster
Not able to deal with multiple requests at the same time

What is method swizzling?
It is a process we can change the functionality of a method at runtime.
//
//Solid Principle in iOS —— other technologies for
Deffination— solid principle are set of design principles that can help you create a maintainable and scalable software..

More principles applies— solid

1-> Single Responsibility Principle:
Means that classes should be handled one specific functionality— like networkManager-DataStoreManager

2-> Open/Closed Principle:
Classes structs should be open for extension but closed for modification

3-> Liskov Substitution Principle:
Means that subclasses should be able to be used interchangeably with their parent class. Like uiviewcontroller ki sub clan banaye loginViewController that loginViewController is changeable.

4-> Interface segregation principle:

Means that developed code by using different different protocol like loader

5-> Dependency inversion protocol:
Menas your app should not believe dependency injection that believe dependency inversion by using protocol inject.