Q1:-Explain what is Xcode?

Sol: Xcode is Apple's integrated development environment (IDE) that you use to build apps for Apple products including the iPad, iPhone, Apple Watch, and Mac. Xcode provides tools to manage your entire development workflow—from creating your app, to testing, optimizing, and submitting it to the App Store.

Q2:-Explain how you can add frameworks in the Xcode project?

Sol: The Steps to add frameworks are:

1. Select "Show the Project navigator"
2. Right-click on the project folder you wish to add the framework to.
3. Select 'Add Files to "YourProjectName"'
4. Browse to the framework - generally under /Developer/SDKs/MacOSXversion.sdk/System/Library/Frameworks
5. Select the one you want.
6. Select "Add"
7. It will appear in both the project navigator where you want it, as well as in the "Link Binary With Libraries" area of the "Build Phases" pane of your target.

Q3:- Explain what is the difference between Xcode, Cocoa and Objective C?

Sol:

Xcode is the integrated development environment (IDE)—the application—that developers use to write software for iOS and/or OS X. It includes the editor, the build system (determining what to build to produce the desired target), and quite a few other things.Xcode includes the Clang compiler, which turns code written in Objective-C, C, and a few other languages into executable code. Most error messages come from Clang, and Xcode relies heavily on it for search indexing, syntax highlighting, and name completion of Objective-C code.

Objective-C is the main language that developers write such software in. They may write bits of it in pure C, use C++ or combine it with Objective-C (producing Objective-C++), or write some or all of the program in another language entirely, such as MacRuby, Java (with j2objc), or C# (with MonoTouch).

Cocoa and Cocoa Touch are application frameworks. Each one is a suite of many individual frameworks (libraries stored in folders named MyFirstFramework), such as:

* Foundation (both): General object-oriented utilities, including NSString, NSURL, NSFileManager, etc.
* Core Foundation (both): Mostly the same thing, but different, mainly in having a C-based API rather than Objective-C
* Application Kit (Mac): Application and windowing framework
* UIKit (iOS): Application and windowing framework
* Core Graphics (both): Drawing
* Core Animation (both): Sprite-management and animation framework

(and many, many more where they came from, especially on the Mac)

i) You can use a framework besides Cocoa or Cocoa Touch, but you'll have to have some Cocoa/Cocoa Touch code, at some level, in order for your application to talk to the system, receive events, draw, etc. Frameworks that specifically target Mac and/or iOS will include the relevant wrappers for you, at varying levels of quality.

ii) You can use a language besides Objective-C, but you'll have lots of syntactic awkwardness, as most of Cocoa and Cocoa Touch are designed for Objective-C's unique syntactic features (particularly selectors).

iii) You can use an editor besides Xcode, and you can even use a build system besides Xcode (such as make), but Apple's App Stores expect apps built a particular way (especially with regard to code signing), so unless you're not targeting the App Stores, any other build system has to replicate what Xcode does or risk its users getting rejections.

Q4:- What is the shortcut to open the “Code Snippet Library” in Xcode?

Sol: You can access the Code Snippet Library from the Xcode Source Editor by choosing View, Utilities, Show Code Snippet Library, or by pressing Control+Option+Command+2. The library appears in the Utility area and is represented by the { } icon

------refer to assets------

1- code snippet,

2- view contents of code snippet

Q5:- Mention what are the build phases available in Xcode?

Sol: The Build phases available in Xcode are:

1. Dependencies
2. Compile Sources
3. Link Binary with Libraries
4. copy Bundle Resources

Q6:- Explain how app delegate is declared by Xcode project templates?

Sol: App delegate is declared as a subclass of UIResponder by Xcode project templates. If the UIApplication object does not handle an event, it dispatches the event to your app delegate for processing.The UIResponder class defines an interface for objects that respond to and handle events.

Q7:- Explain how you define variables in Swift language?

Sol: A variable declaration tells the compiler where and how much to create the storage for the variable. Before you use variables, you must declare them using var keyword as follows −

var variableName = <initial value>

The following example shows how to declare a variable in Swift −

var varA = 42

print(varA)

When we run the above program using playground, we get the following result −

Output - 42

Q8:- What is interface builder?

Sol: The Interface Builder editor within Xcode makes it simple to design a full user interface without writing any code. Simply drag and drop windows, buttons, text fields, and other objects onto the design canvas to create a functioning user interface.

Because Cocoa and Cocoa Touch are built using the Model-View-Controller pattern, it is easy to independently design your interfaces, separate from their implementations. User interfaces are actually archived Cocoa or Cocoa Touch objects (saved as .nib files), and macOS and iOS will dynamically create the connection between UI and code when the app is run.

Q9:- What is the difference between AppDelegate and SceneDelegate?

Sol: AppDelegate is responsible for handling application-level events, like app launch and AppDelegate is the main entry of the app and it is the place where many logics and app states will be handled.It is the place where application launch and apps foreground and background logics are handled.

The SceneDelegate is responsible for scene lifecycle events like scene creation, destruction and state restoration of a UISceneSession.SceneDelegate takes up some responsibilities from AppDelegate. In particular related to UIWindow from AppDelegate is now UIScene in SceneDelegate. An app can have more than one scene which mostly handles application interface and app content. So, the SceneDelegate is responsible for what’s displayed on the screen in terma of UI and data.

Q10:- Which function is called when the app launches from a suspended state?

Sol: The functions that will get called are :

* applicationWillEnterForeground
* applicationWillResignActive

Q11:- Which framework is needed to render UI elements in Playground?

Sol: UIKit framework is needed to render UI elements in Playground.UIkit is a lightweight and modular front-end framework for developing fast and powerful web interfaces.

Q12:- Create an app with a login interface which should have two text fields for email, password where the password is a secure field and a button.

Sol: The solution file is attached with this document.Please refer to it.

Q13:- List devices and their resolutions, screen size has iOS as their OS.

Sol:

| **Device** | **Native Resolution (Pixels)** | **UIKit Size (Points)** | **Native Scale factor** | **UIKit Scale factor** |
| --- | --- | --- | --- | --- |
| iPhone X | 1125 x 2436 | 375 x 812 | 3.0 | 3.0 |
| iPhone 8 Plus | 1080 x 1920 | 414 x 736 | 2.608 | 3.0 |
| iPhone 8 | 750 x 1334 | 375 x 667 | 2.0 | 2.0 |
| iPhone 7 Plus | 1080 x 1920 | 414 x 736 | 2.608 | 3.0 |
| iPhone 6s Plus | 1080 x 1920 | 375 x 667 | 2.608 | 3.0 |
| iPhone 6 Plus | 1080 x 1920 | 375 x 667 | 2.608 | 3.0 |
| iPhone 7 | 750 x 1334 | 375 x 667 | 2.0 | 2.0 |
| iPhone 6s | 750 x 1334 | 375 x 667 | 2.0 | 2.0 |
| iPhone 6 | 750 x 1334 | 375 x 667 | 2.0 | 2.0 |
| iPhone SE | 640 x 1136 | 320 x 568 | 2.0 | 2.0 |
| iPad Pro 12.9-inch (2nd generation) | 2048 x 2732 | 1024 x 1366 | 2.0 | 2.0 |
| iPad Pro 10.5-inch | 2224 x 1668 | 1112 x 834 | 2.0 | 2.0 |
| iPad Pro (12.9-inch) | 2048 x 2732 | 1024 x 1366 | 2.0 | 2.0 |
| iPad Pro (9.7-inch) | 1536 x 2048 | 768 x 1024 | 2.0 | 2.0 |
| iPad Air 2 | 1536 x 2048 | 768 x 1024 | 2.0 | 2.0 |
| iPad Mini 4 | 1536 x 2048 | 768 x 1024 | 2.0 | 2.0 |