1. **Latest iOS Version - 10.3.1**

2. **Xcode Version - 8.3.2**

3. **reuseidentifier -tableview and collection view**

Ans : The reuseIdentifier is used to indicate that a cell can be re-used in a UITableView. For example when the cell looks the same, but has different content. The UITableView will maintain an internal cache of UITableViewCell's with the reuseIdentifier and allow them to be re-used when dequeueReusableCellWithIdentifier: is called. By re-using table cell's the scroll performance of the tableview is better because new views do not need to be created.

4. **App life cycle**

Ans: User taps the app icon the application main() class is called then it will call the application delegate. In appDelegate the Applicationdidfinishlaunching (first initialization). This will activate the application and the ApplicationdidBecomeActive and ApplicationWillEnterForeground method of Appdelegate is called.
If we switch the app or go to background then the deactivation of app starts by calling ApplicationwillResignActive . Then ApplicationdidEnterBackground is called. Then the app will move to suspended state. If the app is killed then ApplicationwillTerminate is called.

5. **atomic and nonatomic**

Ans : Atomic and non-atomic refers to whether the setters/getters for a property will atomically read and write values to the property. When the atomic keyword is used on a property, any access to it will be "synchronized". Therefore a call to the getter will be guaranteed to return a valid value, however this does come with a small performance penalty. Hence in some situations nonatomic is used to provide faster access to a property, but there is a chance of a race condition causing the property to be nil under rare circumstances (when a value is being set from another thread and the old value was released from memory but the new value hasn't yet been fully assigned to the location in memory for the property).

6. difference between copy and retain
Retaining an object means the retain count increases by one. This means the instance of the object will be kept in memory until it's retain count drops to zero. The property will store a reference to this instance and will share the same instance with anyone else who retained it too. Copy means the object will be cloned with duplicate values. It is not shared with any one else.

7. category
A category is a way of adding additional methods to a class without extending it. It is often used to add a collection of related methods. A common use case is to add additional methods to built in classes in the Cocoa frameworks. For example adding async download methods to the UIImage class.

8. protocol

A protocol is similar to an interface from Java. It defines a list of required and optional methods that a class must/can implement if it adopts the protocol. Any class can implement a protocol and other classes can then send messages to that class based on the protocol methods without it knowing the type of the class.

```
@protocol MyCustomDataSource
- (NSUInteger)numberOfRecords;
- (NSDictionary *)recordAtIndex:(NSUInteger)index;
@optional
- (NSString *)titleForRecordAtIndex:(NSUInteger)index;
@end
```
A common use case is providing a DataSource for UITableView or UICollectionView.

9. KVC and KVO?

KVC 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.

Let's say there is a property name on a class:

```
@property (nonatomic, copy) NSString *name;
```

We can access it using KVC:

```
NSString *n = [object valueForKey:@"name"]
```

And we can modify it's value by sending it the message:

```
[object setValue:@"Mary" forKey:@“name”]
```

10. What mechanisms does iOS provide to support multi-threading?

- NSThread creates a new low-level thread which can be started by calling the startmethod.

```
NSThread* myThread = [[NSThread alloc] initWithTarget:self
                                        selector:@selector(myThreadMainMethod:)
                                        object:nil];
[myThread start];
```

- NSOperationQueue allows a pool of threads to be created and used to execute NSOperations in parallel. NSOperations can also be run on the main thread by asking NSOperationQueue for the mainQueue.

```
NSOperationQueue* myQueue = [[NSOperationQueue alloc] init];
[myQueue addOperation:anOperation];
[myQueue addOperationWithBlock:^{
   /* Do something. */
}];
```

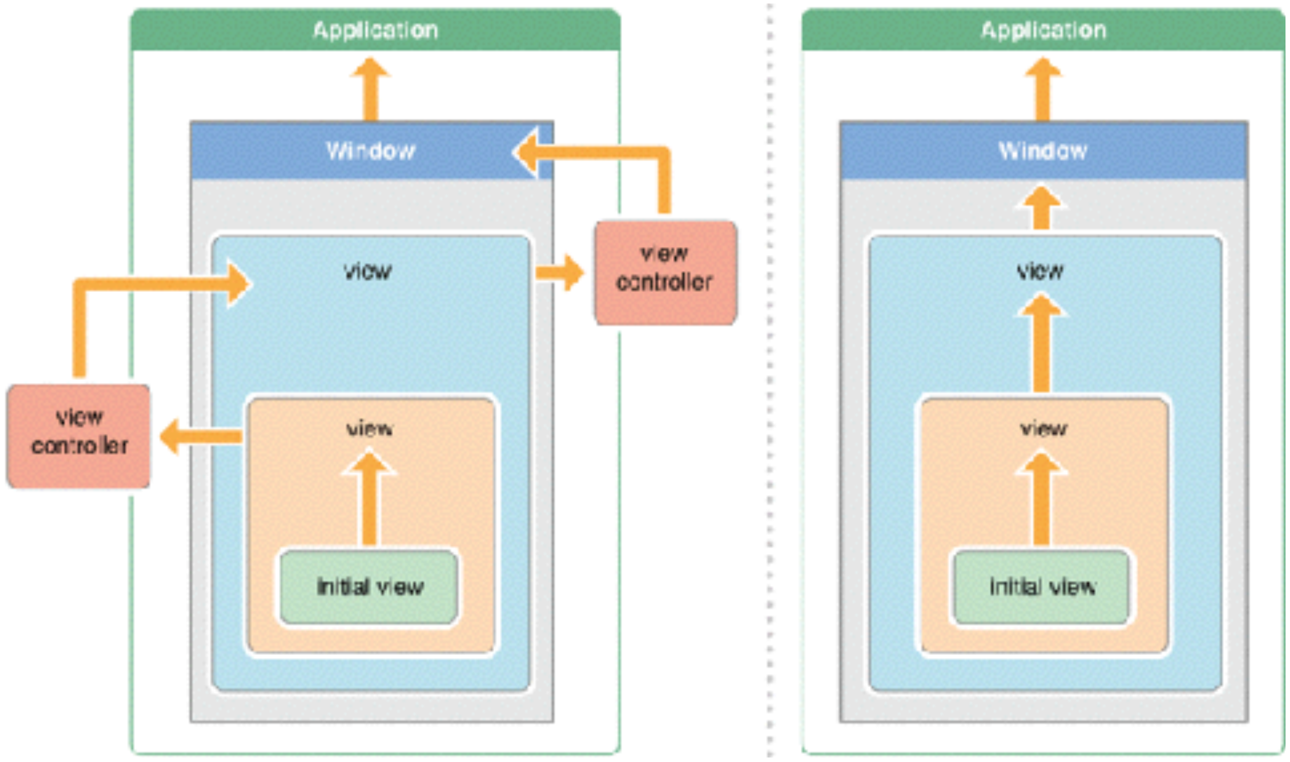
- GCD or Grand Central Dispatch is a modern feature of Objective-C that provides a rich set of methods and API's to use in order to support common multi-threading tasks. GCD provides a way to queue tasks for dispatch on either the main thread, a concurrent queue (tasks are run in parallel) or a serial queue (tasks are run in FIFO order).

```
dispatch_queue_t myQueue =
dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT,
0);
dispatch_async(myQueue, ^{
    printf("Do some work here.\n");
});
```

11. What is the Responder Chain?

When an event happens in a view, for example a touch event, the view will fire the event to a chain of UIResponder objects associated with the UIView. The first UIResponder is the UIView itself, if it does not handle the event then it continues up the chain to until UIResponder handles the event. The chain will include UIViewControllers, parent UIViews and their associated UIViewControllers, if none of those handle the event then the UIWindow is asked if it can handle it and finally if that doesn't handle the event then the UIApplicationDelegate is asked.

If you get the opportunity to draw this one out, it's worth doing to impress the interviewer:



12. What's the difference between using a delegate and notification?

Both are used for sending values and messages to interested parties. A delegate is for one-to-one communication and is a pattern promoted by Apple. In delegation the class raising events will have a property for the delegate and will typically expect it to implement some protocol. The delegating class can then call the _delegate_s protocol methods.

Notification allows a class to broadcast events across the entire application to any interested parties. The broadcasting class doesn't need to know anything about the listeners for this event, therefore notification is very useful in helping to decouple components in an application.

```
[NSNotificationCenter defaultCenter]
        postNotificationName:@"TestNotification"
        object:self];
```

13. What's your preference when writing UI's? Xib files, Storyboards or programmatic UIView?

There's no right or wrong answer to this, but it's great way of seeing if you understand the benefits and challenges with each approach. Here's the common answers I hear:

- Storyboard's and Xib's are great for quickly producing UI's that match a design spec. They are also really easy for product managers to visually see how far along a screen is.
- Storyboard's are also great at representing a flow through an application and allowing a high-level visualization of an entire application.
- Storyboard's drawbacks are that in a team environment they are difficult to work on collaboratively because they're a single file and merge's become difficult to manage.
- Storyboards and Xib files can also suffer from duplication and become difficult to update. For example if all button's need to look

identical and suddenly need a color change, then it can be a long/ difficult process to do this across storyboards and xibs.

- Programmatically constructing UIView's can be verbose and tedious, but it can allow for greater control and also easier separation and sharing of code. They can also be more easily unit tested.

Most developers will propose a combination of all 3 where it makes sense to share code, then re-usable UIViews or Xib files.

14. What is MVC and how is it implemented in iOS?

MVC stands for Model, View, Controller. It is a design pattern that defines how to separate out logic when implementing user interfaces. In iOS, Apple provides UIView as a base class for all _View_s, UIViewController is provided to support the Controller which can listen to events in a View and update the View when data changes. The Model represents data in an application and can be implemented using any NSObject, including data collections like NSArray and NSDictionary.

Some of the pitfalls that people hit are bloated UIViewController and not separating out code into classes beyond the MVC format. I'd highly recommend reading up on some solutions to this:

- https://www.objc.io/issues/1-view-controllers/lighter-view-controllers/
- https://speakerdeck.com/trianglecocoa/unburdened-viewcontrollers-by-jay-thrash
- https://programmers.stackexchange.com/questions/177668/how-to-avoid-big-and-clumsy-uitableviewcontroller-on-ios

In terms of alternatives, this is pretty open ended. The most common alternative is MVVM using ReactiveCocoa, but others include VIPER and using Functional Reactive code.

15. What is AutoLayout? What does it mean when a constraint is "broken" by iOS?

AutoLayout is way of laying out UIViews using a set of constraints that specify the location and size based relative to other views or based on explicit values. AutoLayout makes it easier to design screens that resize and layout out their components better based on the size and orientation of a screen. _Constraint_s include:

- setting the horizontal/vertical distance between 2 views
- setting the height/width to be a ratio relative to a different view
- a width/height/spacing can be an explicit static value

Sometimes constraints conflict with each other. For example imagine a UIView which has 2 height constraints: one says make the UIView 200px high, and the second says make the height twice the height of a button. If the iOS runtime can not satisfy both of these constraints then it has to pick only one. The other is then reported as being "broken" by iOS.

Check this one also:

http://www.sm-cloud.com/ios-interview-questions-for-senior-developers/#3howmemorymanagementishandledonios