iPhone Basic Interview Questions and Answers for Developers

iPhone Basic Interview Questions and Answers for Developers

If you are preparing for iPhone technical interview, you must go through following iPhone interview questions and answers which are very basic for an iPhone developer and you should know all these basic iPhone concepts. Following iPhone interview questions and answers cover basic iPhone concepts like iPhone OS, SDK, Architecture, MVC design pattern, COCOA, COCOA Touch, Objective C, Shallow and Deep Copy in Objective C, difference between implementing a category and inheritance, Apple push notification service, When to use NSMutableArray and when to use NSArray, difference between frame and bounds, UIWindow object, Garbage collector in iPhone, nonatomic, @synthesize, delegate methods of MKMapView, types of parser and lot more...Lets have look on these iPhone basic interview questions and answers.

1. What is iPhone OS?

iPhone OS runs on iPhone and iPod touch devices. Hardware devices are managed by iPhone OS and provides the technologies needed for implementing native applications on the phone. The OS ships with several system applications such as Mail, Safari, Phone, which provide standard services to the user.

2. What is iPhone SDK?

iPhone SDK is available with tools and interfaces needed for developing, installing and running custom native applications.

SDK means **software development kit**. It contains

Xcode IDE,
Instruments,
iPhone simulator,
Interface Builder,Frameworks…

3. What is iPhone Architecture?

It is similar to MacOS X architecture. It acts as an intermediary between the iPhone and iPod hardware an the appearing applications on the screen. The user created applications never interact directly with the appropriate drivers, which protects the user applications from changes to the hardware.

iphone architecture:
---->iOS consists of a number of different software layers, each of which provides programming frameworks for the development of applications that run on top of the underlying hardware.

iOS Layers

1.The Cocoa Touch Layer:

2.The iOS Media Layer

3.iOS Audio Support.

4.The iOS Core Services Layer

5.The iOS Core OS Layer:

The Cocoa Touch Layer:

---->The Cocoa Touch layer sits at the top of the iOS stack and contains the frameworks that are most commonly used by iPhoneapplication developers. Cocoa Touch is primarily written in Objective-C, is based on the standard Mac OS X Cocoa API (as found on Apple desktop and laptop computers) and has been extended and modified to meet the needs of the iPhone hardware Ca130569$
The Cocoa Touch layer provides the following frameworks for iPhone app development:
UIKit Framework (UIKit.framework): ▪ User interface creation and management (text fields, buttons, labels, colours, fonts etc)

- Multitasking
- Wireless Printing
- Data protection via encryption
- Cut, copy, and paste functionality
- Push notification in conjunction with Push Notification Service
- Local notifications (a mechanism whereby an application running in the background can gain the user's attention)
- Accelerometer, battery, proximity sensor, camera and photo library interaction
- Touch screen gesture recognition

Map Kit Framework (MapKit.framework):to get a map of a specific area or to generate driving directions to get you to your intended destination. The Map Kit framework provides a programming interface which enables you to build map based capabilities into your own applications. This allows you to, amongst other things, display scrollable maps for any location, display the map corresponding to the current geographical location of the device and annotate the map in a variety of ways.

Push Notification Service: The Push Notification Service allows applications

to notify users of an event even when the application is not currently running on the device

Message UI Framework (MessageUI.framework): The Message UI framework provides everything you need to allow users to compose and send email messages from within your application

iAd Framework (iAd.framework): The purpose of the iAd Framework is to allow developers to include banner advertising within their applications. All advertisements are served by Apple's own ad service.

Twitter Framework (Twitter.framework): The Twitter Framework allows Twitter integration to be added to applications. The framework operates in conjunction the Accounts Framework to gain access to the user's Twitter account information.

2.The iOS Media Layer

The role of the Media layer is to provide iOS with audio, video, animation and graphics capabilities

Core Graphics Framework (CoreGraphics.framework):

Quartz Core Framework (QuartzCore.framework): The purpose of the Quartz Core framework is to provide animation capabilities on the iPhone.

OpenGL ES framework (OpenGLES.framework): For many years the industry standard for high performance 2D and 3D graphics drawing has been OpenGL.

NewsstandKit Framework (NewsstandKit.framework): The Newsstand application is a new feature of iOS 5 and is intended as a central location for users to gain access to newspapers and magazines.

3.iOS Audio Support

AV Foundation framework (AVFoundation.framework): An Objective-C based framework designed to allow the playback, recording and management of audio content.

Media Player Framework (MediaPlayer.framework): The iOS Media Player framework is able to play video in .mov, .mp4, .m4v, and .3gp formats at a variety of compression standards, resolutions and frame rates.

4.The iOS Core Services Layer

Address Book Framework (AddressBook.framework): The Address Book framework provides programmatic access to the iPhone Address Book contact database allowing applications to retrieve and modify contact entries.

CFNetwork Framework (CFNetwork.framework): The CFNetwork framework provides a C-based interface to the TCP/IP networking protocol stack and low level access to BSD sockets. This enables application code to

be written that works with HTTP, FTP and Domain Name servers and to establish secure and encrypted connections using Secure Sockets Layer (SSL) or Transport Layer Security (TLS).

Core Data Framework (CoreData.framework):

Core Foundation Framework (CoreFoundation.framework): The Core Foundation framework is a C-based Framework which provides basic functionality such as data types, string manipulation, raw block data management, URL manipulation, threads and run loops, date and times, basic XML manipulation and port and socket communication.

Foundation Framework (Foundation.framework):

Core Location Framework (CoreLocation.framework): The Core Location framework allows you to obtain the current geographical location of the device (latitude, longitude and altitude) and compass readings from with your own applications.,This will either be based on GPS readings, Wi-Fi network data or cell tower triangulation (or some combination of the three).

SQLite library: Allows for a lightweight, SQL based database to be created and manipulated from within your iPhone application.

5.The iOS Core OS Layer:

The Core OS Layer occupies the bottom position of the iOS stack and, as such, sits directly on top of the device hardware.

Security Framework (Security.framework): The iOS Security framework provides all the security interfaces you would expect to find on a device that can connect to external networks including certificates, public and private keys, trust policies, keychains, encryption, digests and Hash-based Message Authentication Code (HMAC).

Accelerate Framework (Accelerate.framework):

External Accessory Framework (ExternalAccessory.framework): Provides the ability to interrogate and communicate with external accessories connected physically to the iPhone via the 30-pin dock connector or wirelessly via Bluetooth.

Project templates:

Begin by making sure that the Application option located beneath iOS is selected. The main panel contains a list of templates available to use as the basis for an application. The options available are as follows:

- **Master-Detail Application** – Used to create a list based application. Selecting an item from a master list displays a detail view corresponding to the selection. The template then provides a Back button to return to the list. You may have seen a similar technique used

for news based applications, whereby selecting an item from a list of headlines displays the content of the corresponding news article. When used for an iPad based application this template implements a basic split-view configuration.

- **Open GL Game** – As discussed in iPhone iOS 5 Architecture and SDK Frameworks, the OpenGL ES framework provides an API for developing advanced graphics drawing and animation capabilities. The Open GL ES Game template creates a basic application containing an Open GL ES view upon which to draw and manipulate graphics and a timer object.
- **Page-based Application** – Creates a template project using the page view controller designed to allow views to be transitioned by turning pages on the screen.
- **Tabbed Application** – Creates a template application with a tab bar. The tab bar typically appears across the bottom of the device display and can be programmed to contain items which, when selected, change the main display to different views. The iPhone's built-in Phone user interface, for example, uses a tab bar to allow the user to move between favorites, contacts, keypad and voicemail.
- **Utility Application** – Creates a template consisting of a two sided view. For an example of a utility application in action, load up the standard iPhone weather application. Pressing the blue info button flips the view to the configuration page. Selecting Done rotates the view back to the main screen.
- **Single View Application** – Creates a basic template for an application containing a single view and corresponding view controller.
- **Empty Application** – This most basic of templates creates only a window and a delegate. If none of the above templates match your requirements then this is the option to take.

--->**Automatic Reference Counting** is a new feature included with the Objective-C compiler which removes much of the responsibility from the developer for releasing objects when they are no longer needed.
Accessing the Storyboard: Upon creation of the new project, Xcode will have created what appears to be the usual collection of files for a single view application. Instead of a NIB file for the initial view controller, however, Xcode has created a storyboard file named *MainStoryboard.storyboard*.

4. What is MVC ? MVC Architecture of iPhone App.

Here are the reasons why we should use the MVC (Model View Controller)design pattern.

1. MVC is reusable: When the problems occurs, there is no need to invent a

new solution, we just have to follow the pattern and adopt it as necessary.

2. MVC is expressive: By using the MVC design pattern our application becomes more expressive.

1). Model: The model object knows about all the data that need to be displayed. It is model who is aware about all the operations that can be applied to transform that object. It only represents the data of an application. The model represents enterprise data and the business rules that govern access to and updates of this data. Model is not aware about the presentation data and how that data will be displayed to the browser.

2). View: The view represents the presentation of the application. The view object refers to the model. It uses the query methods of the model to obtain the contents and renders it. The view is not dependent on the application logic. It remains same if there is any modification in the business logic. In other words, we can say that it is the responsibility of the of the view's to maintain the consistency in its presentation when the model changes.

3). Controller: Whenever the user sends a request for something then it always go through the controller. The controller is responsible for intercepting the requests from view and passes it to the model for the appropriate action. After the action has been taken on the data, the controller is responsible for directing the appropriate view to the user. In GUIs, the views and the controllers often work very closely together.

5. What are the ways to store data locally on iPhone device?

We store data locally in device through:

1. Plist.
2. NSUserDefaults.
3. SQLite.
4. CoreData.

6. Difference between COCOA, COCOA touch and objective C?

Objective C is a dynamic programming language - a bit like C++ and a bit like Java.

Cocoa is the application framework for Mac OS X. Cocoa Touch is the application framework for iPhone and iPod Touch - very similar to Cocoa.

Cocoa is commonly referred to as the combination of the Foundation and AppKit frameworks, while Cocoa Touch is the combination of the Foundation and UIKit frameworks. Cocoa and Cocoa Touch sit on top of other collections

of frameworks to create the API stacks. The other layers are Media, Core Services and Core OS. The main difference between Cocoa and Cocoa touch is that the UI classes and APIs aren't the same as Mac OS X, so instead of NSTextField, you have UITextField. Many of the classes share the same functionality and can be ported quite easily by simply changing the class name, though most will require some more changes, but usually nothing too heavy. There are also some differences between the Foundation frameworks in Cocoa and Cocoa Touch, most commonly missing classes, eg, Cocoa has NSHost and Cocoa Touch doesn't.

7. Difference between shallow copy and deep copy?

Shallow copy is also known as address copy. In this process you only copy address not actual data while in deep copy you copy data. Suppose there are two objects A and B. A is pointing to a different array while B is pointing to different array. Now what I will do is following to do shallow copy.?Char *A = {‘a’,’b’,’c’};?Char *B = {‘x’,’y’,’z’};?B = A;?Now B is pointing is at same location where A pointer is pointing.Both A and B in this case sharing same data. if change is made both will get altered value of data.Advantage is that coping process is very fast and is independent of size of array.while in deep copy data is also copied. This process is slow but Both A and B have their own copies and changes made to any copy, other will copy will not be affected.

8. What is advantage of categories? What is difference between implementing a category and inheritance?

You can add method to existing class even to that class whose source is not available to you. You can extend functionality of a class without subclassing. You can split implementation in multiple classes. While in Inheritance you subclass from parent class and extend its functionality.

9. Flow of push notification?

Ans. Your web server sends message (device token + payload) to Apple push notification service (APNS) , then APNS routes this message to device whose device token specified in notification.

10. What is polymorphism?

This is very famous question and every interviewer asks this. Few people say polymorphism means multiple forms and they start giving example of draw function which is right to some extent but interviewer is looking for more detailed answer. Ability of base class pointer to call function from derived class at runtime is called polymorphism. For example, there is super class human and there are two subclasses software engineer and hardware engineer. Now super class human can hold reference to any of subclass

because software engineer is kind of human. Suppose there is speak function in super class and every subclass has also speak function. So at runtime, super class reference is pointing to whatever subclass, speak function will be called of that class. I hope I am able to make you understand.

11. When to use NSMutableArray and when to use NSArray?

Ans. Normally we use mutable version of array where data in the array will change. For example, you are passing a array to function and that function will add some elements to that array or will remove some elements from array, then you will select NSMutableArray. When you don't want to change you data, then you store it into NSArray. For example, the country names you will put into NSArray so that no one can accidentally modify it.

12. How is the app delegate is declared by Xcode project templates?

App delegate is declared as a subclass of UIResponder by Xcode project templates.

13. What is the purpose of UIWindow object?

The presentation of one or more views on a screen is coordinated by UIWindow object.

14. Whats the difference between frame and bounds?

The frame of a view is the rectangle, expressed as a location (x,y) and size (width,height) relative to the superview it is contained within. The bounds of a view is the rectangle, expressed as a location (x,y) and size (width,height) relative to its own coordinate system (0,0).

15. What is @interface?

It's a keyword used to declare the Class.

16. What is @implementation?

It's a keyword used to define the Class.

17. Garbage collector in iPhone?

iOS has got the ARC (Automated reference counting). Objective C does not have a garbage collector rather it uses the reference counting algorithm to manage the memory. This was the developers task until Apple launched iOS 5.0. Again if you are targeting iOS 4.0 or earlier , ARC is no more a choice for you.

18. What is delegate?

Delegate is an object that handles the events happening on an object. To do that delegate has to follow a protocol specifying the task it is going to handle.

19. What is @synthesize?

We use @synthesize to generate getters and setters automatically from compiler. We declare properties and then generate getter and setter method by using @synthesize.

20. What is nonatomic?

nonatomic and atomic are related to multithreading environment . If a property has an attribute as “nonatomic” that means multiple threads can modify that property concurrently. If the attribute is “atomic”, the threads would be given access atomically. So “Atomic” is thread safe while “nonatomic” is thread unsafe. Atomic drastically hampers the performance so until and unless not needed you should never go for atomic attribute. ‘nonatomic ’ will do in most of the cases.

21. What are the delegate methods of MKMapView?

Firstly you have added the storeKit framework in your Xcode project then define the protocol as <MKMapviewDelegate> in .h file.

- (void)mapView:(MKMapView *)mapView regionWillChangeAnimated:(BOOL)animated;
- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;
- (void)mapViewWillStartLoadingMap:(MKMapView *)mapView;
- (void)mapViewDidFinishLoadingMap:(MKMapView *)mapView;
- (void)mapViewDidFailLoadingMap:(MKMapView *)mapView withError:(NSError *)error;
- (MKAnnotationView *)mapView:(MKMapView *)mapView viewForAnnotation:(id <MKAnnotation>)annotation;
- (void)mapView:(MKMapView *)mapView didAddAnnotationViews:(NSArray *)views;

22. What are the important delegate methods of NSXML parser?

-DidStartElement
-FoundCharecters
-DidEndElement
-FoundError

23. What is @dynamic and any place where it is used?
It tells compiler that getter and setter are not implemented by the class but by some other class.
May be super class or child class.
Example – Core Data.
- The Managed object classes have properties defined by using @dynamic.

24. Types of parsers?
There are various parsers available to parse an XML in iOS app development. You can choose either from iOS SDK provided by Apple or from third party libraries. Before discussing about various options, I want to highlight difference between XML parsers. Generally two types of parsers are used in XML parsing: SAX and DOM. Let's have a look at their brief description.
SAX parser- This parser is based on some notifying methods. These methods are notified as the SAX parser moves on the XML document. During the parsing activity, developer is responsible to construct required object and keep track of state of the XML parser. Some examples of this parser are as follows-
NSXMLParser - It is written in Objective-C and provided by Apple through its iOS SDK.
libxml2 - It is based on C language API and also provided by Apple through its iOS SDK. It supports both SAX and DOM parser.
DOM parser- This parser parse the complete document in a single go and converts it into a specific structured object. We can create XPath query for a particular element from this structured object. Some examples of this parser are as follows-
TBXML- It's a lightweight XML parser designed in such a way to consume very low memory. It's a good choice for an XML as well as that have the fixed structure.
TouchXML - It is another DOM parser. It is also read only but does not support XPath queries.
KissXML - It is based on TouchXML parser. It supports editing and writing XML unlike TouchXML parser.
TinyXML - It is very small DOM parser and based on C language API. It supports editing and writing XML documents like KissXML parser but does not support XPath queries.
GDataXML - It is developed by Google using Objective-C API. It supports both editing of XML documents and XPath queries.

sandeepkumar

8978310091

***Q: What are all the difference between categories and subclasses? Why should we go to subclasses?**

A:Category is a feature of the Objective-C language that enables you to add methods (interface and implementation) to a class without having to make a subclass. There is no runtime difference—within the scope of your program—between the original methods of the class and the methods added by the category. The methods in the category become part of the class type and are inherited by all the class's subclasses.As with delegation, categories are not a strict adaptation of the Decorator pattern, fulfilling the intent but taking a different path to implementing that intent. The behaviour added by categories is a compile-time artifact, and is not something dynamically acquired. Moreover, categories do not encapsulate an instance of the class being extended.The Cocoa frameworks define numerous categories, most of them informal protocols . Often they use categories to group related methods. You may implement categories in your code to extend classes without subclassing or to group related methods. However, you should be aware of these caveats:

- You cannot add instance variables to the class.
- If you override existing methods of the class, your application may behave unpredictably.

Difference between notifications and delegates

The two differences between delegates and notifications is that delegate is a one-to-one connection between the delegate and the delegating object. Notifications are posted application wide and can be observed by as many objects as needed creating a one-to-many connection Think about this as a broadcast. The second main difference is that delegation can be used to transfer information back from the delegate to the delegating object. Meaning that the delegating object asks the delegate for certain information. Typical example would be the data source of an UITableView. Notifications however are a one way street. The information flows from the posting object to the observing objects. This makes sense because think about the situation where you would have more than one observer and each would give feedback to the posting objects. Which one would be the right one?

1.What is the RAM size in iPhone 4?

512MB

2.Current device resolution?

320X480 for iPhone
1024X720 for iPad

3.What compiler we have used in Objective c?

Compiler for objective c GCC version 4.0

6.What is Xcode?

Xcode is an Integrated Development Environment
IDE acts like an Editor of Objective c language.
We can write, run, build our code in Xcode

7.What is the simulator?

it is a demo to check our applications if we didn't have the original device

8.What are Instruments:

The instrument enables you to dynamically trace and profile the performance of the
Mac os, iPhone, iPad applications.
By using Instruments we can test the performance of the application.

9.what is Interface Builder:

Interface builder is developed in 1988.It is a visual tool that allows you to design your interfaces for iPhone. By using interface builder we can drag and drop the views on interface.

10.Explain about Inspector

It is mainly used for setting the properties of view elements. It contains four sections.

1)Attribute inspector
2)Connection
3)Size of inspector
4)Identity

11.Groups and File sections:

Classes:

In class folder actual coding will be there.
for window based application we have two files.

1)appdelegate.h
2)appdelegate.m

For view based applications we have four files.

1)appdelegate.h
2)appdelegate.m
3)viewcontrol.h
4)viewcontrol.m

12.What are different Layers of IOS

Cocoa Touch Layer

Media Layer

Core Services Layer

Core Services Layer

13.What is optimization?
Optimization is nothing but Making our application more attractively even after completion of the coding

14.What happened if array release?
As such, when the NSArray is released your object will still have a retain count of one and will therefore still be around after the autorelease pool is emptied. (Until of course you release it.

15.How can we use multiple class inheritance in objective c?
In Objective c Multiple inheritance is not supported
for this purpose we use "categories" in Objective c

16.How can u use object for abstract class?
(or) What is Shared Obj?
We can't create object for abstract class
to call abstract class we use Shared obj

17.Garbage collection:

Garbage collection means a memory management system that automatically release memory used by unreferenced variable.

Modern computer languages uses Garbage collection [GC]
Gc is a runtime algorithm that scans the allocated objects in our program
and deallocates the obj that we have loss contact

cocoa under the iphone os does not use GC so cocoa apps must use managed memory

Applications running on this platform should b clean up after use.
since iphone apps run in a memory constraint environment

18.Retain:
It indicates that we can putting the Ownership climb on the object

It increases the Retain count by 1,
If the caller releases the object it doesn't deallocates if it is retained
19.Release:

It decreases the retain count by 1

When we call release we r no longer the owner of the obj

If the the reference count is 0, the object will b Deallocates

The memory is freed it will be used by another obj

20.AUTORELEASEPOOL:

All the released objects are added to the Pool, that pool is nothing but AutoReleasePool

The pool objects are released by keywords release/Drai

21.What is the difference between the Dot notation and using the Square brackets?

The Dot Syntax only for Setter & Getters
button for the Methods

22.Reference counting.

Objective-C's memory management system is called **Reference counting**.

23.How can we deallocates all objects in AutoReleasepool?
difference b/n Drain & Release..?

By Using DRAIN keyword we can de allocate all objects from autoReleasePool

By using Release We can't De allocate all the objects

24.Memory management:

De alloc

The dealloc method is called on an object when it is being removed from memory. This is usually the best time to release references to all of your child instance variables:

```
- (void) dealloc
{
    [caption release];
    [photographer release];
    [super dealloc];
}
```

On the first two lines, we just send release to each of the instance variables. We don't need to use autorelease here, and the standard release is a bit faster.

Memory leak.

The last line is very important. We have to send the message [super dealloc] to ask the superclass to do its cleanup. If we don't do this, the object will not be removed, which is a **Memory leak.**

Reference counting.
Objective-C's memory management system is called **Reference counting**. All you have to do is keep track of your references, and the runtime does the actual freeing of memory.
In simplest terms, you alloc an object, maybe retain it at some point, then send one release for each alloc/retain you sent. So if you used alloc once and then retain once, you need to release twice.
You create an object with alloc or copy, send it a release or autorelease message at the end of the function. If you create an object any other wise nothing.
Dealloc

It is called when the viewcontroller object is deallocated, which it will be when the retain count drops to zero.

25.What is Framework?

Collection of classes is nothing but a Frame Work
Frame work is a library.It contains all collection of predefined classes,functions,protocols....

26.What is Protocol?

A protocol is a collection of methods that any class can choose to implement. This can be useful when you are creating family of similar classes that all need to communicate with a common control class.

27.What is Category?

A category is a way of adding new methods to the all instances of existing class with out modifying the class. The use of category is extending the NSString class to add functionality.

28.What is Property:

By using property what ever the data are stored in array,string,… that values are used in any where in that program.Property contains the data and synthesize retrieve the data.

29.What are the property declarations/attributes?

atomic,
nonatomic,
readonly,retain, copy.

30.What are the types of Accessory methods?

We have 2 types of ac…. methods:

i.Getter method:

Getter method is an accessor method that retrieves the value of an instance variable.

ii.Setter method:

Setter method is an accessor method that Sets the value of an instance variable.

31.What is a Selector?

Selector can either be a name of method Or message to an object.

32.What is Delegate?

It is an object that usually read to some events in another object

A Delegate allows one object to send messages to another object when an Event happens.

An object directly to carryout an action by another object.

33.What is an App Delegate..?

It is necessary in any app.

It is a controller which takes care critical events while Starting Running and Editing applications

34.iPhone Paths:

iPhone contains two paths.

1.Resource path:

By using this path we can retrieve the data from resource path.But we can not add the data.

2.Modify path

By using this path we can add,modify,delete,retrieve the data.

35.Accelerometer:

The accelerometer allows the device to detect the orientation of the device and adapts the content to suit the new orientation.

36.Header file:

Header files are preprocessor directories.

in Obj c By using #include,#import we can import the header files.

37.Diff between self and super:

Self:

self is a variable that refers to the object of present class.

Super:

Super refers to the same variable.

38.Resources:

In resources we can add the outside contents like images, videos, audio files.

It contains info.plist file.plist file having entire project information.

39.Products:

The final output present in product folder and extension is .app

Intially .app is in red color. once if you can build the app then the .app is created.

40.Files owner window:

It contains copy of nib files.

It is always 1st icon in any .nib file.

41.What Enum?

It is a user defined data type Iit improves speed of execution.

By default it assigns range from 0,1……

42.Forward declarations:

@class,@protocols these are forward declarations to avoid cyclic dependencies.

@class:

This directive to make a forward reference to another class and improve the .h file.

@protocols:

These are two types.

1)Formal protocol 2)Informal protocol

Formal protocol:

It is a set of methods that must be implemented in any conformed class.

Informal protocal:

In informal protocol developer group methods by app fields.

…… IMP FRAMEWORKS…..

43.Foundation framework

It provides foundation classes for objective-c.
such as NSobject,basic data types.

44.UIKit framework:

It provides fundamental objects for managing the UI applications.

45.core location framework:

It provides location based information using combination of GPS,cell ID,hi-fi networks.

46.Mapkit framework: It provides an Embedded map interface for your application.

47.Media Player framework:

It provides facilities for playing audio and video files.

48.Core Graphics framework:

It provides c-based APIS for 2d rendering based on quartz drawing engine.

49.AVFoundation framework:

It provides low level c APIS for audio recording and playback as well as managing audio hardware.

50.Interface:

In interface section that declares the methods and instance variables of class.

51.Implementation:

In implementation section that actuvally defines the class.

52.What kind of design pattern we are using?

We are using model view controller

Model:

This layer manipulates the coding part of the application.

View:

This layer manipulates the design part of the application.

Controller:

This layer acts as bridge between model and view controller.

53.What is Id?

It is a data type.

Id is a generic c-type used by obj-c to refer to any object.

54.NSFile mager

By using NSfile manager we can compare, add, delete and update the data in database.

SQLite

55.SQLIte

It is one of the database to store the large number of data.

KeySQLite methods:

1)SQLite3_open():

This function opens specific database file.if the database file does not already exists it is created.

2)SQLite3_close():

This function closes if previously opened database file.

3)SQLite3_prepare_v2():

This function prepares a SQL statement ready for execution.

4)SQLite3_step():

This function executes a SQL statement previously prepared by using sqlite_prepare_v2.

5)SQLite_column_<type>():

This function returns a datafield from the results of a SQL retrieval operation. where <type> is replaced by datatype of the data to be extractes.

6)SQLite3_finalize():

This function deletes a previously prepared SQL statement from memory.

7)SQLite3_exec():

This function combines the functionality of SQLite3_prepare_v2(), SQLite3_step(),SQLite3_finalize(). into a single function call.

56.NSBundle mainbundle:

It is used to pick the data from resources.

57.UI application:

Every iPhone has UI application.it is the starting point of the every application.

It is responsible for initializing and display your application on UI Window and also responsible for loading your application.

58.Controllers in iPhone:

By using controllers we can move from one view to another view.

ex: Pushviewcontroller,
Popviewcontroller,
Presentmodelviewcontroller,
Dismissmodelviewcontroller.

59.How can we deploy our app?

By using standard edition worth 99$ we can deploy our app

60.How can we Localize our app?

For localization have to use current language

For exam to make a delete button
in English: Delete
in Japanese lan: Aurem
in Spanish: Delat

in this way we can localize our app

61.Why We use Authentication?

For the security purpose we use authentication

62.When keyboard appears, hides some part of UI for that what shall we do?

keyboard will appear only in some views like "Textfield", "Textviews" etc but not in all the views

for display the keyboard we have to use "becomeFirstResponder"..
for hides the keyboard we have to use "resignFirstResponder"
ex:

```
[textField becomeFirstResponder];
[textField resignFirstResponder];
```

63.becomeFirstResponder?

For display the keyboard we have to use "becomeFirstResponder"..

ex:[textField becomeFirstResponder];

64.What is awakeFromNib, becomeFirstResponder?

Activity should be allocated from loading xib file, it should be not nil in awakeFromNib
And awakeFromNib is called when all file owners outlets and properties are set (including *view* property).
So it makes sense that *viewDidLoad* is called earlier than *awakeFromNib*.

65.What is Context?

Context refers to the conditions in which something exists or occurs.

One is connected with databases, persistence layers, graphics and such beasts
where you need some notion of a ‘scope’, ‘connection’ or a ‘state’.
For example when saving data into a database
you usually need to open the database and then save some DB ‘handle’
You will refer to in subsequent operations.
There can be many different connections and thus many different ‘handles’.
In other words there can be many DB *contexts*

66.Difference Between nil & NiL

Both nil, Nil return a NULL pointer (Zero pointer)

but
nil is specific to objects (e.g., *id*)
and *Nil* is specific to class pointers.

X [for the NULL pointers in Objective c We have to use :nil

in the same way
for the NULL pointers in C,C++ We have to use :NiL
NiL is just like NULL]

67.What happens when we invoke a method on a nil pointer

It returns 0, nil, a structure filled with 0s,

68.Multiple class inheritance in objective c?

In Objective c Multiple inheritance is not supported
for this purpose we use "categories" in Objective c

Categories: A category allows you to add methods to an existing class… Categories are a powerful feature that allows you to extend the functionality of existing classes without subclassing.The declaration of a category interface looks very much like a class interface declaration Except the category name is listed within parentheses after the class name and the superclass isn't mentioned. Unless its methods don't access any instance variables of the class,

70.Web services

"Web service" as "a software system designed to support interoperable machine-to-machine interaction over a network

Web services are typically application programming interfaces (API) or web APIs that are accessed via Hypertext Transfer Protocol and executed on a remote system hosting the requested services

Web Services delegate Methods

- (void)connection:(NSURLConnection *)connection didReceiveResponse:(NSURLResponse *)response

- (void)connection:(NSURLConnection *)connection didReceiveData:(NSData*)data
- (void)connectionDidFinishLoading:(NSURLConnection *)connection
- (void)connection:(NSURLConnection *)connection didFailWithError:(NSError *)error

SAX vs. DOM

Before we begin, I wanted to make sure everyone is aware of the most important difference between XML parsers:
whether the parser is a SAX or a DOM parser.

* A SAX parser is one where your code is notified as the parser walks through the XML tree,
and you are responsible for keeping track of state and constructing any objects you might want to keep track of the data as the parser marches through.

* A DOM parser reads the entire document and builds up an in-memory representation that you can query for different elements.

Often, you can even construct XPath queries to pull out particular pieces.

72.What is XML parsing?

XML parsing is the transfer of data from xml page to our application the data which is in term of elements.

73.The Most Popular XML Parsers for the iPhone

In my research, here is what seemed to me to be the most popular XML Parsers for the iPhone, and a brief description of each one:

* NSXMLParser is a SAX parser included by default with the iPhone SDK. It is written in Objective-C and is quite straightforward to use, but perhaps not quite as easy as the DOM model.

* libxml2 is an Open Source library that is included by default with the iPhone SDK. It is a C-based API, so is a bit more work to use than NSXML. The library supports both DOM and SAX processing. The libxml2 SAX processor is especially cool, as it has a unique feature of being able to parse the data as it is being read. For example, you could be reading a large XML document from the network and displaying data that youíre reading for it to the user while youíre still downloading.

* TBXML is a lightweight DOM XML parser designed to be as quick as possible while consuming few memory resources. It saves time by not performing validation, not supporting XPath, and by being read-only i.e. you can read XML with it, but you can't then modify the XML and write it back out again.

* TouchXML is an NSXML style DOM XML parser for the iPhone. Like TBXML, it is also read-only, but unlike TBXML it does support XPath.

* KissXML is another NSSXML style DOM XML parser for the iPhone, actually based on TouchXML. The main difference is KissXML also supports editing and writing XML as well as reading.

* TinyXML is a small C-based DOM XML parser that consists of just four

C files and two headers.
It supports both reading and writing XML documents, but it does not support XPath on its own. However, you can use a related library TinyXPath for that.

* GDataXML is yet another NSXML style DOM XML parser for the iPhone, developed by Google as part of their Objective-C client library.
Consisting of just a M file and a header, it supports both reading and writing XML documents and XPath queries.

Ok, now let us start comparing all these libraries!

74.What r the Delegate Methods for XML Parsing?
There are mainly 3Types of Delegate methods for XML Parsing
i. did start element
ii. did end element
iii. found characters

75.What is Picker view..?
Picker view lets the user select an item from a list.

76.What is Table View?
It is commonly used to show list of data in the form of Records

77.WEb Services?
Web services are application programming interfaces [API]
or Web API's that are accessed via Hyper Text protocol and excited on a remote system
hosting the requested services

78.Difference between HTML & XML?
XML was designed to transfer and store the data
XML was to carry data not to display the data
XML is designed to be self description

HTML was designed to Display the data

79.Core DATA:
The ways to persist data on the iPhone,
Core Data is the best one to use for non-trivial data storage.
It can reduce the memory overhead of your app,
increase responsiveness, and save you from writing a lot of boilerplate code.

Apple has finally ported Core Data to the iPhone which means using SQLite in

an application

80.FILE SHARING:
The Data in Plists is existed in the form of Files..
Retrieve these file from plist to out application
is nothing but a File Sharing

81.Threads:
Threads are one of several technologies that make it possible to execute multiple code paths concurrently inside a single application.

83.Shallow copy , Deep Copy:

Shallow copies duplicate as little as possible.

A shallow copy of a collection is a copy of the collection structure, not the elements.

With a shallow copy, two collections now share the individual elements.

Deep copies duplicate everything. A deep copy of a collection is two collections with all of the elements in the original collection duplicated.

Shallow copy: Copies the member values from one object into another.

Deep Copy: Copies the member values from one object into another.
Any pointer objects are duplicated and Deep Copied.

84.What happens if you remove the object from the array, and you try to use it?

Our application crashes

85.Diff b/w viewWillAppear & viewDidLoad

viewWillAPpear is called each time the view becomes visible.
viewDidLoad is only called when the view is initialized.

viewDidLoad being called once when the view is loaded (actually it may be called more than once if it was released and created again)
ViewWillAppear is called each time the view is going to appear (so put state updates here)

viewDidLoad is things you have to do once. viewWillAppear gets called every time the view appears.

You should do things that you only have to do once in viewDidLoad - like setting your UILabel texts. However, you may want to

Modify a specific part of the view every time the user gets to view it, e.g. the iPod application scrolls the lyrics back to

the top every time you go to the "Now Playing" view.

86.When didReceiveMemory gets called

It is called when the system is low on memory.

didReceiveMemoryWarning will be called before an out-of-memory crash. Not other crashes.
If you handle the warning properly and free up memory, then you can avoid the out-of-memory condition and not crash.

The purpose of didReceiveMemoryWarning is to give you a chance to free memory or pop views to avoid a crash.

You can manually trigger a memory warning in the simulator under the Hardware menu. Highly recommend doing this
To flush out problems.
Instruments helps you debug leaks (though not all of them) - it's not really that useful for crashes.

90.What is the difference between KVO & KVC

The views use KVC to update the model,

and they use KVO to watch for changes in the model.

113.What is multiple inheritance does objective c supports for it?

In object-oriented programming, the ability of a class to have more than one superclass—to inherit from different sources and thus combine separately-defined behaviors in a single class. Objective-C doesn’t support multiple inheritance.

1.What if Life Cycle of an iphone application?
A.When user touches the icon of the application main() gets called by the system. main() creates the autoreleasepool and starts the application with UIApplicationMain object.

UIApplicationMain creates the instance of the UIApplication which actually starts the application.

UIApplication loads the main nib file and sets the application. Then UIApplication forwards the events to interface elements.

For terminating the application user taps home button, UIApplication tells the delegate that application is terminating. And so UIApplication gets exited , then main gets exited. This way the process ends.

This is the way iPhone applications works.

2.Diff'e bet'n Frame and Bounds?

A.The frame is the view's location in its superview, using superview's coordinate system.

The bounds is its location and size in its own coordinate system.

If you are getting the same results for both properties, it means that the view fills its superview, and both views have (0, 0) as the origin. Try changing the frame, and you will see it move to different positions within its superview.

3.Protocol?
A. Is a collection of methods that any class can choose to implement.

4.categories

A category is a methodology in objective-c it helps us to extend an existing class without modifying the original source code for class and without subclassing it

5.difference between Mutable and Immutable?

Mutable means you can change its contents later but when you mark any object immutable, it means once they are initialized, their values cannot be changed. For example, NSArray, NSString values cannot be changed after initialized.

6.What is StoryBoard?

With Storyboards, all screens are stored in a single file. This gives you a conceptual overview of the visual representation for the app and shows you how the screens are connected. Xcode provides a built-in editor to layout the

Storyboards.

- .storyboard is essentially one single file for all your screens in the app and it shows the flow of the screens. You can add segues/transitions between screens, this way. So, this minimizes the boilerplate code required to manage multiple screens.
- Minimizes the overall no. of files in an app.

7.how to relate core data & sqlite?
A.Core Data can serialize the objects into XML and SQ-Lite for storage..

8.what is XML Parsing?
A.xml parsing is the transfer of data from xml page to our application
the data which is in term of elements.

10.what is meant by NSXML parser , and where you have used in your app?

it is one of the class it is used when we parse the data from xml

11.which is best in XML and JSON and why?
A.JSON because its a lightweight process

12.what are the web services available?
A.XML,JSON

13. what are XML parsing delegates used in your app?

Sax Parser : Simple API of XML Parse node to node, using top-down traversing, parse without storing xml, Faster compared to Dom Manipulating of node like insertion or deletion is allowed. Needs SAXParserFactory

Dom Parser : Document Object Model Stores entire xml in memory before processing, traverse in any direction, Manipulating of node like insertion or deletion is NOT allowed. Needs DocumentBuilderFactory

Pull Parser: It provides more control and speed from the above two.

SAX will generally use much less memory. And you only handle the events you want, e.g. startElement(), startDocument(), endElement(), etc. etc., and thus you will only save specific data you want. DOM will load the entire XML file into memory.

did start element

did end element

Found characters

14.what is SQLite?

A.SQlite having 9- Properties:
1.SQLite-3 Statement
2.SQLite-3 Prepare:Table creation
3.SqLite-3 FInalize
4.Sqlite-3 Close
5.SqLite-3 Open:
6. SqLite-3 Create: to create table
7. Sqlite-3 Step:Retriving data
8.Sqlite-3 Exec: inserting data

15.push notifications

Push notification allows an app to notify user with new messages or events without the need to actually open the application

steps followed in push notifications

ios device to apns(Apple push notification server)

apns to ios device

ios to app

16.deploying app in app store

A:

- Obtaining your iOS Distribution Certificate
- Create and download your iOS Distribution Provisioning Profile for App Store Distribution
- Creating and Downloading a Distribution Provisioning Profile for Ad Hoc Distribution

- Building your Application with Xcode for Distribution
- Verifying a Successful Distribution Build
- Updating your Application

Hiteshi Technologies,Indore
1.brief description about your profile?
2.challenges that you face in your apps?
3.what is XML Parsing?
A.xml parsing is the transfer of data from xml page to our application the data which is in term of elements.
4.which is best in XML and JSON and why?
A.JSON because its a lightweight process
5.what are the web services available?
A.XML,JSON,Plist
6.what are the web services that you use in your applications?
7.facebook integration (social networking)?
8.SAX Vs. DOM parser?

Tekneekllc,kolkata
1.what is GCD?

A. Grand Central Dispatch(like multi-threading)

2.what is paypal Integration?
3.what is NSObserver?
4.what is NSCoding?
5.what is Push Notification?
A. Its like Alert messages its also called as remote notifications.
6.what is SQLite?
A.SQlite having 9- Properties:
1.SQLite-3 Statement
2.SQLite-3 Prepare:Table creation
3.SqLite-3 FInalize
4.Sqlite-3 Close
5.SqLite-3 Open:
6. SqLite-3 Create: to create table
7. Sqlite-3 Step:Retriving data
8.Sqlite-3 Exec: inserting data

7.what are the queries that you use in SQLite
8.how you will fetch the data from SQLite?

appstudioz,noida
1.what is push notification?what steps to be followed to include push

notifications in your app?
2.what is inapp purchase?
3.what is difference between auto resizing and auto layout?
4.Difference between protocols & categories?
5.what is use of storyboards?
6.how to relate core data & sqlite?
A.Core Data can serialize the objects into XML and SQ-Lite for storage..
7.Protocol?
A. Is a collection of methods that any class can choose to implemen….
8.Inapp purchaset
9,model view controller

1.What if Life Cycle of an iphone application?
A.When user touches the icon of the application main() gets called by the system. main() creates the autoreleasepool and starts the application with UIApplicationMain object.

UIApplicationMain creates the instance of the UIApplication which actually starts the application.

UIApplication loads the main nib file and sets the application. Then UIApplication forwards the events to interface elements.

For terminating the application user taps home button, UIApplication tells the delegate that application is terminating. And so UIApplication gets exited , then main gets exited. This way the process ends.

This is the way iPhone applications works.

2.What Is Protocol?

A. Protocol is collection of methods that any class can chose to implement

3.Diff’e bet’n Frame and Bounds?

A.The frame is the view's location in its superview, using superview's coordinate system.

The bounds is its location and size in its own coordinate system.

If you are getting the same results for both properties, it means that the

view fills its superview, and both views have (0, 0) as the origin. Try changing the frame, and you will see it move to different positions within its superview.

4.what is pushnotification?

5.what are the web services and which type of parsing tech are used?

XML JSON

6.What is ARC

Teknowledge-Mobile-Studio (kolkata)

1.protocol

Protocol is collection of methods that any class can chose to implement

2.categories

A category is a methodology in objective -c it helps us to extend an existing class without modifying the original source code for class and without subclassing it

3.ARC

4.when the object released when we use ARC

5.private and public Variables

6.sqlite

7.deploying app in app store

A:

- Obtaining your iOS Distribution Certificate
- Create and download your iOS Distribution Provisioning Profile for App Store Distribution

- Creating and Downloading a Distribution Provisioning Profile for Ad Hoc Distribution
- Building your Application with Xcode for Distribution
- Verifying a Successful Distribution Build
- Updating your Application

8.push notifications

Push notification allows an app to notify user with new messages or events without the need to actually open the application

steps followed in push notifications

ios device to apns(Apple push notification server)

apns to ios device

ios to app

9.lifecycle of iphone application?

10.difference between Mutable and Immutable?

11.tell me about inapppurchase?

12.tell me about table view?

SIgnity Software Solutions

1. What is meant by parsing?

getting the data from the server

2. what are XML parsing delegates used in your app?

http://www.tekritisoftware.com/xml-parsers-iOS-app-development

Sax Parser : Simple API of XML Parse node to node, using top-down traversing, parse without storing xml, Faster compared to Dom Manipulating of node like insertion or deletion is allowed. Needs SAXParserFactory

Dom Parser : Document Object Model Stores entire xml in memory before

processing, traverse in any direction, Manipulating of node like insertion or deletion is NOT allowed. Needs DocumentBuilderFactory

Pull Parser: It provides more control and speed from the above two.

SAX will generally use much less memory. And you only handle the events you want, e.g. startElement(), startDocument(), endElement(), etc. etc., and thus you will only save specific data you want. DOM will load the entire XML file into memory.

did start element

did end element

found characters

3. what are JSON parsing delegates are u used in your app?

4. what are the dataBases used in your applications?

SQLite,coredata

5. what is meant by pushnotification?

Push notification allows an app to notify user with new messages or events without the need to actually open the application

6.what is meant by NSXML parser , and where you have used in your app?

it is one of the class it is used when we parse the data from xml

7. what is the difference between protocol and categeory?

8. what is difference between copy and retain?

AppStudioz -Noida and three companies

1. Push Notification?

2.APNS server?

A.Apple Push Notification Server

3.what is meant by DataEnd Post?

4 Can you please tell me about core data ?

5.XML parser And JSON Parser?

6.What is meant by In APP purchase?

7.Have you developed any chat application? tell me in brief?

8. what is the difference between protocol and categeory?

9.difference between Mutable and Immutable?

Mutable means you can change its contents later but when you mark any object immutable, it means once they are initialized, their values cannot be changed. For example, NSArray, NSString values cannot be changed after initialized.

10.what is meant strong?

A1 Techonlogies - Mohali,Punjab

1. What is meant by dynamic keyword?(you can get these word at @synthesize)

2.What is meant by Category?

3.What is meant by Notifications?

4.what is basic difference between HTML and XML?

5.What delegates are used in XML parser?

6.what is meant by autorelease?

7.what is procedure involved to convert statements in to SQLite Database?

8.What is meant by Data End Post?

9.What is basic difference between Notifications and Delegates?

10.What is meant by NSOperationQueue?

11.What is StoryBoard?

With Storyboards, all screens are stored in a single file. This gives you a conceptual overview of the visual representation for the app and shows you how the screens are connected. Xcode provides a built-in editor to layout the Storyboards.

- .storyboard is essentially one single file for all your screens in the app and it shows the flow of the screens. You can add segues/transitions between screens, this way. So, this minimizes the boilerplate code required to manage multiple screens.
- 2. Minimizes the overall no. of files in an app.

Lets Nature

1.tell me about u r self?

2.what is ARC?

3.What is Synthesized

4.What is Static,retain?

5.What is NSCoder?

6.What is singleton class?

A singleton is a special kind of class where only one instance of the class exists for the current process

7.what is diffe between .h & .m files?

Appspeak -Kolkatta

1.What is shallow Copy and deep copy?

2.What is KVC & KVO?

3.What is NSNotifications?

4.What is atomic and Nonatomic?

5.What is data source?

Likipe Digital AB,**Kolkata**

1.explain briefly about mapkit

2.how to find the distance between 2 places

3.(**V.IMP******)what is your place of your company in your city?

DIETCODE,GOA

1.what is the difference between ios 5 and ios 6?

2.difference between stack and heap?

3.output of the code given below

NSNumber *a=@”6”;

NSNumber *b=@”3”;

NSNumber *c=@”a*b”;

NSLog(@”%@”,c);

4.NSarray consists of 3 objects how to add 4th object is it possible?

Hiteshi Technologies:(Indore)

1.Tell me about yourself?

2.Tell about brief description about your applications?

3.What are web services you have used in your application?

4.what is difference between XML and JSon?

5.How to manage memory in Objective-C

6.What is Arc and any drawback in ARC?

CIS(Indore)

1.what is the difference between asynchronous and synchronous downloading?

2.how will you request for web service?

3.what is GET and POST?

4.what is a web service ?

5.what is delegation?

6.what is push notification?

7.what is in app purchase?

OPENXCELL(ahmedabad)

1.what is interface?

2.what is interface and implementation blocks?

3.what is forward declaration?

4.what is dictionary?

5.what is GCD?

6.is it possible to update UI In background?

No,We can’t UI must always be updated from the main thread

7.what is local notification?

Local notifications are ways for an application that isn’t running in the foreground to let its users know it has information for them.The information could be a message, an impending calendar event, or new data on a remote server

8.what is payload in push notification?

Each push notification includes a payload. The payload contains information about how the system should alert the user as well as any custom data you provide. The maximum size allowed for a notification payload is 256 bytes.

9.Difference between delegate and protocol?

10.what is NSNotificationCenter?

An NSNotificationCenter object provides a mechanism for broadcasting information within a program.Objects register with a notification center to receive notifications (NSNotification objects) using the

addObserver:selector:name:object: or addObserverForName:object:queue:usingBlock: methods. Each invocation of this method specifies a set of notifications

11.what is core data stack?

A stack contains all the Core Data components you need to fetch, create, and manipulate managed objects

12.difference between core data and SQLite?

13.what is NSFileFormatting?

Brain Wire - Mumbai

1. Describe Mobile Development career ?

2. Explain OOP concepts ?

3.Life Cycle of ViewController ?

4. what are the delegate methods in MapKit ?

5. Explain JSON Parsing ?

6. How to Store data in SQLite ?

7. IOS , XCode current versions you are using ?

8. Which FrameWorks you are using in your Project?

9. which type of libraries using your Application ?

10. what are the challenges you are facing in your application ?

11. Explain AutoLayout ?

12. what are the Web Services are use ?

13. what is CoreData ?

14. which library are used for JSON Parsing ?

15. Base64 & NSData

KOLKATTA

1. Why you should choose iPhone ? Any Reason ?

2. Tell me about Object Oriented Programming ?

3. What is Objective-C ?

4. What are the Difference Between Protocol & Categeory ?

5. Tell me about Polymorphism ?

6. What kind of Frameworks used in your Applications ?

7. what are the Web Services that you use in your applications?

8. What is XML ?

9. What is GCD ?

TULI e SERVICES PVT LTD.

Kolkata, India

difference between XML and JSON?

advantages of JSON?

what is synthesize?

What is property?

What is the lifecycle of Iphone Application?

Difference between interface and delegate?

What is ARC? and advantages?

What is retain, assign?

What is push notification?

how do you integrate email in your application?

UITableView delegate methods?

Iphone and ipad screen sizes?

difference between retina display and normal display?

Version of X-Code?

Atomic and non-atomic?

Make 24 using four zeros?

Ans: (0!+0!+0!+0!)!=(1+1+1+1)!=4!=1*2*3*4=24.

default frameworks involved while creating new project?

Ans: foundation.framework, UIKit.framework, CoreGraphics.framework

INSPEERO Mumbai

Tell me about your self?

most difficulties aspects which you faced while developing your applications

what is coredata?

what is singleton?give a real time example for it?

what is a category?

what are the design patterns

what is a notification and its use ?

what is a delegate

what is custom delegate?

how to store data in ios?

disadvantages of coredata compared to sqlite?

brief me about social frame work?

suppose i am having 300 images and i need to download it in when the app is background,how can we handle this process in ios?

what are the web services used in ios?and brief about them?

SPARSH COMMUNICATIONS Hyderabad

tell me about yourself?

what is a category?

what is a delegate?

what is a protocol?

difference between sqlite and coredata?

suppose i am one application in which i am i have stored data and i have uninstalled it and after some period i will again install the same application then the stored previously will like that itself then how you can handle this by using coding in ios?

E2Infosystems (Chennai)

what is InAppParchage?

difference b/w nsarray and namutablearray ?

syntax for imageview?

how can you commit app is forGround and BackGround?

Unicorn Infotech

1. Cocoa Touch UIKit/Foundation framework.

2. Core graphics.

3. Media Kit (Audio and Video).

4. Con-Currency (multitasking).

5. iOS Sandbox environment.
6. In-App purchases.

7. Apple Push Notification Services and local notifications.

8. Navigation based applications.

9. Automatic Reference Counting (ARC).

10. Web services (Using RESTFUL and SOAP) ASIHTTP Request library.

11. Blocks.

12. Data Base management (Core Data and SQLITE3).

13. Animations (different types of layers and usage).

14. Managing backward compatibility and user data when upgrading the app complete knowledge on major features introduced from iOS-4.2 to iOS-7

Navigators Software Pvt Ltd (Kolkata)

1. What is Property

2. What is Synthesized

3. What is keyword Dynamic

4. Difference B/N Core Data & SQLite

5. Difference B/N Delegate & DataSource

6. What is Protocol

7. Difference B/N Instance Method & Class Method

8. What is Category

9. How can you create a Custom Cell in Objective-C

10. How can you Reuse Cell in TableView

11. Can i used two Table Views in One View

12. What is In-App Purchase

MOBICASA

1) What is persistent store coordinator?

2) Asynchronous and synchronous?

3)What is KVO?

4)What is Post Notification?

5) lifecycle of Objective-C?

6) What is Local Database? Ans: SQLITE.

Basic Questions in IOS & objective-c

1-How would you create your own custom view?
By Subclassing the UIView class.
2-Whats fast enumeration?
Fast enumeration is a language feature that allows you to enumerate over the contents of a collection. (Your code will also run faster because the internal implementation reduces message send overhead and increases pipelining potential.)
3-Whats a struct?
A struct is a special C data type that encapsulates other pieces of data into a single cohesive unit. Like an object, but built into C.
4-What are mutable and immutable types in Objective C?
Mutable means you can change its contents later but when you mark any object immutable, it means once they are initialized, their values cannot be changed. For example, NSArray, NSString values cannot be changed after initialized.
5-Explain retain counts.

Retain counts are the way in which memory is managed in Objective-C. When you create an object, it has a retain count of 1. When you send an object a retain message, its retain count is incremented by 1. When you send an object a release message, its retain count is decremented by 1. When you send an object a autorelease message, its retain count is decremented by 1 at some stage in the future. If an object's retain count is reduced to 0, it is deallocated.

6-Whats the difference between frame and bounds?

The frame of a view is the rectangle, expressed as a location (x,y) and size (width,height) relative to the superview it is contained within. The bounds of a view is the rectangle, expressed as a location (x,y) and size (width,height) relative to its own coordinate system (0,0).

7-Is a delegate retained?

No, the delegate is never retained! Ever!

8-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

9- What is dynamic**?**

You use the @dynamic keyword to tell the compiler that you will fulfill the API contract implied by a property either by providing method implementations directly or at runtime using other mechanisms such as dynamic loading of code or dynamic method resolution. It suppresses the warnings that the compiler would otherwise generate if it can't find suitable implementations. You should use it only if you know that the methods will be available at runtime

10-If I call performSelector:withObject:afterDelay: – is the object retained?

Yes, the object is retained. It creates a timer that calls a selector on the current threads run loop. It may not be 100% precise time-wise as it attempts to dequeue the message from the run loop and perform the selector.

11-Can you explain what happens when you call autorelease on an object?

When you send an object a autorelease message, its retain count is decremented by 1 at some stage in the future. The object is added to an autorelease pool on the current thread. The main thread loop creates an autorelease pool at the beginning of the function, and release it at the end. This establishes a pool for the lifetime of the task. However, this also means that any autoreleased objects created during the lifetime of the task are not disposed of until the task completes. This may lead to the task's memory footprint increasing unnecessarily. You can also consider creating pools with a narrower scope or use NSOperationQueue with it's own autorelease pool. (Also important – You only release or autorelease objects you own.)

12-Whats the NSCoder class used for?

NSCoder is an abstractClass which represents a stream of data. They are used in Archiving and Unarchiving objects. NSCoder objects are usually used in a method that is being implemented so that the class conforms to the protocol. (which has something like encodeObject and decodeObject methods

in them).

13-Whats an NSOperationQueue and how/would you use it?

The NSOperationQueue class regulates the execution of a set of NSOperation objects. An operation queue is generally used to perform some asynchronous operations on a background thread so as not to block the main thread.

14-Explain the correct way to manage Outlets memory

Create them as properties in the header that are retained. In the viewDidUnload set the outlets to nil(i.e self.outlet = nil). Finally in dealloc make sure to release the outlet.

15-Is the delegate for a CAAnimation retained?

Yes it is!! This is one of the rare exceptions to memory management rules.

16-What happens when the following code executes?

Ball *ball = [[[[Ball alloc] init] autorelease] autorelease];

It will crash because it's added twice to the autorelease pool and when it it dequeued the autorelease pool calls release more than once.

17-Explain the difference between NSOperationQueue concurrent and non-concurrent.

In the context of an NSOperation object, which runs in an NSOperationQueue, the terms concurrent and non-concurrent do not necessarily refer to the side-by-side execution of threads. Instead, a non-concurrent operation is one that executes using the environment that is provided for it while a concurrent operation is responsible for setting up its own execution environment.

18-Implement your own synthesized methods for the property NSString *title.

Well you would want to implement the getter and setter for the title object. Something like this: view source print?

```
- (NSString*) title // Getter method
{
return title;
}
- (void) setTitle: (NSString*) newTitle //Setter method
{
if (newTitle != title)
{
[title release];
title = [newTitle retain]; // Or copy, depending on your needs.
}
}
```

19-Implement the following methods: retain, release, autorelease.

```
-(id)retain
{
NSIncrementExtraRefCount(self);
return self;
}
-(void)release
{
```

```
if(NSDecrementExtraRefCountWasZero(self))
{
NSDeallocateObject(self);
}
}
-(id)autorelease
{ // Add the object to the autorelease pool
[NSAutoreleasePool addObject:self];
return self
```

20-What are the App states. Explain them?

- **Not running State**: The app has not been launched or was running but was terminated by the system.
- **Inactive state:** The app is running in the foreground but is currently not receiving events. (It may be executing other code though.) An app usually stays in this state only briefly as it transitions to a different state. The only time it stays inactive for any period of time is when the user locks the screen or the system prompts the user to respond to some event, such as an incoming phone call or SMS message.
- **Active state**: The app is running in the foreground and is receiving events. This is the normal mode for foreground apps.
- **Background state**: The app is in the background and executing code. Most apps enter this state briefly on their way to being suspended. However, an app that requests extra execution time may remain in this state for a period of time. In addition, an app being launched directly into the background enters this state instead of the inactive state. For information about how to execute code while in the background, see “Background Execution and Multitasking.”
- **Suspended state:**The app is in the background but is not executing code. The system moves apps to this state automatically and does not notify them before doing so. While suspended, an app remains in memory but does not execute any code. When a low-memory condition occurs, the system may purge suspended apps without notice to make more space for the foreground app.

21-What is Automatic Reference Counting (ARC) ?

ARC is a compiler-level feature that simplifies the process of managing the lifetimes of Objective-C objects. Instead of you having to remember when to retain or release an object, ARC evaluates the lifetime requirements of your objects and automatically inserts the appropriate method calls at compile time.

22-Multitasking support is available from which version?

iOS 4 and above supports multi-tasking and allows apps to remain in the background until they are launched again or until they are terminated.

23-How many bytes we can send to apple push notification server.

256bytes.

24-What is the difference between retain & assign?

Assign creates a reference from one object to another without increasing the source’s retain count.

```
if (_variable != object)
```

```
{
[_variable release];
_variable = nil;
_variable = object;
}
```

Retain creates a reference from one object to another and increases the retain count of the source object.

```
if (_variable != object)
{ [_variable release];
_variable = nil;
_variable = [object retain];
}
```

25-Why do we need to use @Synthesize?

We can use generated code like nonatomic, atmoic, retain without writing any lines of code. We also have getter and setter methods. To use this, you have 2 other ways: @synthesize or @dynamic: @synthesize, compiler will generate the getter and setter automatically for you, @dynamic: you have to write them yourself.@property is really good for memory management, for example: retain.How can you do retain without @property?

```
if (_variable != object)
{
[_variable release];
_variable = nil;
_variable = [object retain];
}
```

How can you use it with @property?self.variable = object;When we are calling the above line, we actually call the setter like [self setVariable:object] and then the generated setter will do its job

26-What is categories in iOS?

A Category is a feature of the Objective-C language that enables you to add methods (interface and implementation) to a class without having to make a subclass. There is no runtime difference—within the scope of your program—between the original methods of the class and the methods added by the category. The methods in the category become part of the class type and are inherited by all the class's subclasses.As with delegation, categories are not a strict adaptation of the Decorator pattern, fulfilling the intent but taking a different path to implementing that intent. The behavior added by categories is a compile-time artifact, and is not something dynamically acquired. Moreover, categories do not encapsulate an instance of the class being extended.

27-What is Delegation in iOS?

Delegation is a design pattern in which one object sends messages to another object—specified as its delegate—to ask for input or to notify it that an event is occurring. Delegation is often used as an alternative to class inheritance to extend the functionality of reusable objects. For example, before a window changes size, it asks its delegate whether the new size is ok. The delegate

replies to the window, telling it that the suggested size is acceptable or suggesting a better size. (For more details on window resizing, see thewindowWillResize:toSize: message.)Delegate methods are typically grouped into a protocol. A protocol is basically just a list of methods. The delegate protocol specifies all the messages an object might send to its delegate. If a class conforms to (or adopts) a protocol, it guarantees that it implements the required methods of a protocol. (Protocols may also include optional methods).In this application, the application object tells its delegate that the main startup routines have finished by sending it theapplicationDidFinishLaunching: message. The delegate is then able to perform additional tasks if it wants.

28-How can we achieve singleton pattern in iOS?

The Singleton design pattern ensures a class only has one instance, and provides a global point of access to it. The class keeps track of its sole instance and ensures that no other instance can be created. Singleton classes are appropriate for situations where it makes sense for a single object to provide access to a global resource.Several Cocoa framework classes are singletons. They include NSFileManager, NSWorkspace, NSApplication, and, in UIKit, UIApplication. A process is limited to one instance of these classes. When a client asks the class for an instance, it gets a shared instance, which is lazily created upon the first request.

29-What is delegate pattern in iOS?

Delegation is a mechanism by which a host object embeds a weak reference (weak in the sense that it's a simple pointer reference, unretained) to another object—its delegate—and periodically sends messages to the delegate when it requires its input for a task. The host object is generally an "off-the-shelf" framework object (such as an NSWindow or NSXMLParserobject) that is seeking to accomplish something, but can only do so in a generic fashion. The delegate, which is almost always an instance of a custom class, acts in coordination with the host object, supplying program-specific behavior at certain points in the task (see Figure 4-3). Thus delegation makes it possible to modify or extend the behavior of another object without the need for subclassing.Refer: delegate pattern

30-What are all the difference between categories and subclasses?Why should we go to subclasses?

Category is a feature of the Objective-C language that enables you to add methods (interface and implementation) to a class without having to make a subclass. There is no runtime difference—within the scope of your program—between the original methods of the class and the methods added by the category. The methods in the category become part of the class type and are inherited by all the class's subclasses.As with delegation, categories are not a strict adaptation of the Decorator pattern, fulfilling the intent but taking a different path to implementing that intent. The behavior added by categories is a compile-time artifact, and is not something dynamically acquired. Moreover, categories do not encapsulate an instance of the class being extended.The Cocoa frameworks define numerous categories, most of

them informal protocols . Often they use categories to group related methods. You may implement categories in your code to extend classes without subclassing or to group related methods. However, you should be aware of these caveats:

- You cannot add instance variables to the class.
- If you override existing methods of the class, your application may behave unpredictably.

31-What is notification in iOS?

The notification mechanism of Cocoa implements one-to-many broadcast of messages based on the Observer pattern. Objects in a program add themselves or other objects to a list of observers of one or more notifications, each of which is identified by a global string (the notification name). The object that wants to notify other objects—the observed object—creates a notification object and posts it to a notification center. The notification center determines the observers of a particular notification and sends the notification to them via a message. The methods invoked by the notification message must conform to a certain single-parameter signature. The parameter of the method is the notification object, which contains the notification name, the observed object, and a dictionary containing any supplemental information.Posting a notification is a synchronous procedure. The posting object doesn't regain control until the notification center has broadcast the notification to all observers. For asynchronous behavior, you can put the notification in a notification queue; control returns immediately to the posting object and the notification center broadcasts the notification when it reaches the top of the queue.Regular notifications—that is, those broadcast by the notification center—are intraprocess only. If you want to broadcast notifications to other processes, you can use the istributed notification center and its related API.

32-What is the difference between delegates and notifications?

We can use notifications for a variety of reasons. For example, you could broadcast a notification to change how user-interface elements display information based on a certain event elsewhere in the program. Or you could use notifications as a way to ensure that objects in a document save their state before the document window is closed. The general purpose of notifications is to inform other objects of program events so they can respond appropriately.But objects receiving notifications can react only after the event has occurred. This is a significant difference from delegation. The delegate is given a chance to reject or modify the operation proposed by the delegating object. Observing objects, on the other hand, cannot directly affect an impending operation.

33-What is posing in iOS?

Objective-C permits a class to **entirely replace another class**within an application. The replacing class is said to "pose as" the target class. All messages sent to the target class are then instead received by the posing class. There are some restrictions on which classes can pose:

- A class may only pose as one of its direct or indirect superclasses
- The posing class must not define any new instance variables which are

absent from the target class (though it may define or override methods).

- No messages must have been sent to the target class prior to the posing.

Posing, similarly to categories, allows **globally augmenting existing classes**. Posing permits two features absent from categories:

- A posing class can call overridden methods through super, thus incorporating the implementation of the target class.
- A posing class can override methods defined in categories.

34-What is atomic and nonatomic? Which one is safer? Which one is default?

You can use this attribute to specify that accessor methods are not atomic. (There is no keyword to denote atomic.)

nonatomic

Specifies that accessors are nonatomic. By default, accessors are atomic.

Properties are atomic by default so that synthesized accessors provide robust access to properties in a multithreaded environment—that is, the value returned from the getter or set via the setter is always fully retrieved or set regardless of what other threads are executing concurrently.

If you specify strong, copy, or retain and do not specifynonatomic, then in a reference-counted environment, a synthesized get accessor for an object property uses a lock and retains and autoreleases the returned value—the implementation will be similar to the following:

```
[_internal lock]; // lock using an object-level lock

id result = [[value retain] autorelease];

[_internal unlock];

return result;
```

If you specify nonatomic, a synthesized accessor for an object property simply returns the value directly.

35-Where can you test Apple iPhone apps if you don't have the device?

iOS Simulator can be used to test mobile applications. Xcode tool that comes along with iOS SDK includes Xcode IDE as well as the iOS Simulator. Xcode also includes all required tools and frameworks for building iOS apps. However, it is strongly recommended to test the app on the real device before publishing it.

36-Which JSON framework is supported by iOS?

SBJson framework is supported by iOS. It is a JSON parser and generator for Objective-C. SBJson provides flexible APIs and additional control that makes JSON handling easier.

37-What are the tools required to develop iOS applications?

iOS development requires Intel-based Macintosh computer and iOS SDK.

38- Name the framework that is used to construct application's user interface for iOS.

A. The UIKit framework is used to develop application's user interface for iOS. UIKit framework provides event handling, drawing model, windows, views, and controls specifically designed for a touch screen interface.

39-Name the application thread from where UIKit classes should be used?

UIKit classes should be used only from an application's main thread. Note: The derived classes of UIResponder and the classes which manipulate application's user interface should be used from application's main thread.

40- Which API is used to write test scripts that help in exercising the application's user interface elements?

UI Automation API is used to automate test procedures. Tests scripts are written in JavaScript to the UI Automation API. This in turn simulates user interaction with the application and returns log information to the host computer.

41-Why an app on iOS device behaves differently when running in foreground than in background?

An application behaves differently when running in foreground than in background because of the limitation of resources on iOS devices.

42- How can an operating system improve battery life while running an app?

An app is notified whenever the operating system moves the apps between foreground and background. The operating system improves battery life while it bounds what your app can do in the background. This also improves the user experience with foreground app.

43-Which framework delivers event to custom object when app is in foreground?

The UIKit infrastructure takes care of delivering events to custom objects. As an app developer, you have to override methods in the appropriate objects to process those events.

44-When an app is said to be in not running state?

An app is said to be in 'not running' state when: - it is not launched. - it gets terminated by the system during running.

45-Assume that your app is running in the foreground but is currently not receiving events. In which sate it would be in?

An app will be in InActive state if it is running in the foreground but is currently not receiving events. An app stays in InActive state only briefly as it transitions to a different state.

46- Give example scenarios when an application goes into InActive state?

An app can get into InActive state when the user locks the screen or the system prompts the user to respond to some event e.g. SMS message, incoming call etc.

47-When an app is said to be in active state?

An app is said to be in active state when it is running in foreground and is

receiving events.

48-Name the app sate which it reaches briefly on its way to being suspended

An app enters background state briefly on its way to being suspended.

49- Assume that an app is not in foreground but is still executing code. In which state will it be in?

Background state.

50-An app is loaded into memory but is not executing any code. In which state will it be in?

An app is said to be in suspended state when it is still in memory but is not executing any code.

51-Assume that system is running low on memory. What can system do for suspended apps?

In case system is running low on memory, the system may purge suspended apps without notice.

52- How can you respond to state transitions on your app?

On state transitions can be responded to state changes in an appropriate way by calling corresponding methods on app's delegate object.

For example: applicationDidBecomeActive method can be used to prepare to run as the foreground app. applicationDidEnterBackground method can be used to execute some code when app is running in the background and may be suspended at any time. applicationWillEnterForeground method can be used to execute some code when your app is moving out of the background applicationWillTerminate method is called when your app is being terminated.

53-List down app's state transitions when it gets launched.

Before the launch of an app, it is said to be in not running state. When an app is launched, it moves to the active or background state, after transitioning briefly through the inactive state.

54-Who calls the main function of you app during the app launch cycle?

During app launching, the system creates a main thread for the app and calls the app's main function on that main thread. The Xcode project's default main function hands over control to the UIKit framework, which takes care of initializing the app before it is run.

55-What is the use of controller object UIApplication?

Controller object UIApplication is used without subclassing to manage the application event loop. It coordinates other high-level app behaviors. It works along with the app delegate object which contains app-level logic.

56-Which object is create by UIApplicationMain function at app launch time?

The app delegate object is created by UIApplicationMain function at app launch time. The app delegate object's main job is to handle state transitions within the app.

57- How is the app delegate is declared by Xcode project templates?

App delegate is declared as a subclass of UIResponder by Xcode project

templates.

58-What happens if IApplication object does not handle an event?

In such case the event will be dispatched to your app delegate for processing.

59- Which app specific objects store the app's content?

Data model objects are app specific objects and store app's content. Apps can also use document objects to manage some or all of their data model objects.

60-Are document objects required for an application? What does they offer?

Document objects are not required but are very useful in grouping data that belongs in a single file or file package.

61- Which object manage the presentation of app's content on the screen?

View controller objects takes care of the presentation of app's content on the screen. A view controller is used to manage a single view along with the collection of subviews. It makes its views visible by installing them in the app's window.

62- Which is the super class of all view controller objects?

UIViewController class. The functionality for loading views, presenting them, rotating them in response to device rotations, and several other standard system behaviors are provided by UIViewController class.

63-What is the purpose of UIWindow object?

The presentation of one or more views on a screen is coordinated by UIWindow object.

64-How do you change the content of your app in order to change the views displayed in the corresponding window?

To change the content of your app, you use a view controller to change the views displayed in the corresponding window. Remember, window itself is never replaced.

65-Define view object.

Views along with controls are used to provide visual representation of the app content. View is an object that draws content in a designated rectangular area and it responds to events within that area.

66-Apart from incorporating views and controls, what else an app can incorporate?

Apart from incorporating views and controls, an app can also incorporate Core Animation layers into its view and control hierarchies.

67- What are layer objects and what do they represent?

Layer objects are data objects which represent visual content. Layer objects are used by views to render their content. Custom layer objects can also be added to the interface to implement complex animations and other types of sophisticated visual effects.

68-What is App Bundle?

When you build your iOS app, Xcode packages it as a bundle. A**bundle** is a directory in the file system that groups related resources together in one place. An iOS app bundle contains the app executable file and supporting resource files such as app icons, image files, and localized content.

69-Define property?
It is used to access instance variables outside of class.
70-Why synthesized is used?
After declaring property we will have to tell compiler instantly by using synthesize directive. This tells the compiler to generate setter and getter methods.
71-What is retaining?
It is reference count for an object.
72- What is webservice?
To get data in form of xml ,by using this we can get data from a server.
73-What is parsing?
To get data from web service we use parsing.
74-which xml parser we use on iphone?
“NSXML” Parser.
75-Which type of parse does iphone support?
“SAX” parser.
76-.Name those classes used to establish connection b/w application to webserver?
(a)NSURL (b)NSURL REQUEST (c)NSURL CONNECTION.
77-Tell the difference between DOM and SAX Parser?
(a)Dom is “documents based parser”. b)SAX is a event driven parser
78-Name three method of NSXML parser.
(1)did start element (2)did end element (3)found character.
79-Tell methods used in NSURLConnection
(1)Connection did receive Response (2)Connection did recevice Datat (3)Connection fail with error (4)Connection did finish loading.
80-.What is json-parser?
JSON(Java script object notation)is a parser used to get data from web Server.
81-.By default which things are in the application?
iPhone applications by default have 3 things 1.main: entry point of application. 2.Appdelegate: perform basic application and functionality. 3.Window: provide uiinterface.
82-Tell me about tab bar controller?
It is used to display the data on the view.
83-Which are the protocols used in table view?
Table view contain two delegate protocols (1) Uitable view data source (2).Uitable view delegate. ui view table view data source three method namely (1)No of sections. (2)No of rows in sections. (3)Cell for row index path row. In ui table view delegate contain (1)Did select row at index path row
84-Name data base used in iPhone?
(1)Sql lite (2)Plist 3)Xml (4)Core Data
85-Tell four frameworks used in iPhone?
(1)Ui kit framework (2)Map kit framework (3)ADI kit framework (4)Core

data framework (5)core foundation framework

86-Tell me about single inheritance in objective-c?

Objective c subclass can derived from a single parent class.It is called “single inheritance”.

87-Tell me about the MVC architecture?

M-model, V-view, C-controller

Main advantage of MVC architecture is to provide “reusability and security” by separating the layer by using MVC architecture.

Model: it is a class model is interact with database.

Controller: controller is used for by getting the data from model and controls the views.

Display the information in views. : View

88-What is the instance methods?

Instance methods are essentially code routines that perform tasks so instances of classes we create methods to get and set the instance variables and to display the current values of these variables.

Declaration of instance method :

– (void)click me: (id)sender;

Void is return type which does not giving any thing here.

Click me is method name.

Id is data type which returns any type of object.

89-What is the class method?

Class methods work at the class level and are common to all instance of a class these methods are specific to the class overall as opposed to working on different instance data encapsulated in each class instance.

@interface class name :ns object

{

}

+(class name *)new alloc:

-(int)total open

90-What is data encapsulation?

Data is contained within objects and is not accessible by any other than via methods defined on the class is called data encapsulation.

91-What is accessor methods?

Accessor methods are methods belonging to a class that allow to get and set the values of instance valuables contained within the class.

92-What is synthesized accessor methods?

Objective-c provides a mechanism that automates the creation of accessor methods that are called synthesized accessor methods that are implemented through use of the @property and @synthesized.

93-How to access the encapsulated data in objective-c?

(a)Data encapsulation encourages the use of methods to +get and set the values of instance variables in a class.

(b)But the developer to want to directly access an instance variable without having to go through an accessor method.

(c) In objective-c syntax for an instance variable is as follow [class instance

variable name]

94-What is dot notation?

Dot notation features introduced into version 2.0 of objective-c. Dot notation involves accessing an instance variable by specifying a class “instance” followed by a “dot” followed in turn by the name of instance variable or property to be accessed.

95-Difference between shallow copy and deep copy?

Shallow copy is also known as address copy. In this process you only copy address not actual data while in deep copy you copy data.

Suppose there are two objects A and B. A is pointing to a different array while B is pointing to different array. Now what I will do is following to do shallow copy. Char *A = {‘a’,’b’,’c’}; Char *B = {‘x’,’y’,’z’}; B = A; Now B is pointing is at same location where A pointer is pointing.Both A and B in this case sharing same data. if change is made both will get altered value of data.Advantage is that coping process is very fast and is independent of size of array.

while in deep copy data is also copied. This process is slow but Both A and B have their own copies and changes made to any copy, other will copy will not be affected.

96-Difference between categories and extensions?

Class extensions are similar to categories. The main difference is that with an extension, the compiler will expect you to implement the methods within your main @implementation, whereas with a category you have a separate @implementation block. So you should pretty much only use an extension at the top of your main .m file (the only place you should care about ivars, incidentally) — it’s meant to be just that, an extension.

97-What are KVO and KVC?

KVC: Normally instance variables are accessed through properties or accessors but KVC gives another way to access variables in form of strings. In this way your class acts like a dictionary and your property name for example “age” becomes key and value that property holds becomes value for that key. For example, you have employee class with name property.

You access property like

NSString age = emp.age;

setting property value.

emp.age = @”20″;

Now how KVC works is like this

[emp valueForKey:@"age"];

[emp setValue:@"25" forKey:@"age"];

KVO : The mechanism through which objects are notified when there is change in any of property is called KVO.

For example, person object is interested in getting notification when accountBalance property is changed in BankAccount object.Toachieve this, Person Object must register as an observer of the BankAccount’s accountBalance property by sending an addObserver:forKeyPath:options:context: message.

98-Can we use two tableview controllers on one view controller?
Yes, we can use two tableviews on the same view controllers and you can differentiate between two by assigning them tags…or you can also check them by comparing their memory addresses.
99-Swap the two variable values without taking third variable?
int x=10; int y=5; x=x+y; NSLog(@"x==> %d",x);
y=x-y; NSLog(@"Y Value==> %d",y);
x=x-y; NSLog(@"x Value==> %d",x);
100-What is push notification?
Imagine, you are looking for a job. You go to software company daily and ask sir "is there any job for me" and they keep on saying no. Your time and money is wasted on each trip.(Pull Request mechanism)
So, one day owner says, if there is any suitable job for you, I will let you know. In this mechanism, your time and money is not wasted. (Push Mechanism)
How it works?
This service is provided by Apple in which rather than pinging server after specific interval for data which is also called pull mechanism, server will send notification to your device that there is new piece of information for you. Request is initiated by server not the device or client.
Flow of push notification
Your web server sends message (device token + payload) to Apple push notification service (APNS) , then APNS routes this message to device whose device token specified in notification.
101-What is polymorphism?
This is very famous question and every interviewer asks this. Few people say polymorphism means multiple forms and they start giving example of draw function which is right to some extent but interviewer is looking for more detailed answer.
Ability of base class pointer to call function from derived class at runtime is called polymorphism.
For example, there is super class human and there are two subclasses software engineer and hardware engineer. Now super class human can hold reference to any of subclass because software engineer is kind of human. Suppose there is speak function in super class and every subclass has also speak function. So at runtime, super class reference is pointing to whatever subclass, speak function will be called of that class. I hope I am able to make you understand.
101-What is responder chain?
Suppose you have a hierarchy of views such like there is superview A which have subview B and B has a subview C. Now you touch on inner most view C. The system will send touch event to subview C for handling this event. If C View does not want to handle this event, this event will be passed to its superview B (next responder). If B also does not want to handle this touch event it will pass on to superview A. All the view which can respond to touch events are called responder chain. A view can also pass its events to

uiviewcontroller. If view controller also does not want to respond to touch event, it is passed to application object which discards this event.

102-Can we use one tableview with two different datasources? How you will achieve this?

Yes. We can conditionally bind tableviews with two different data sources.

103-What is a protocol?

A protocol is a language feature in objective C which provides multiple inheritance in a single inheritance language. Objective C supports two types of protocols:

- Ad hoc protocols called informal protocol
- Compiler protocols called formal protocols

You must create your own autorelease pool as soon as the thread begins executing; otherwise, your application will leak objects

104-Three occasions when you might use your own autorelease pools:

- If you are writing a program that is not based on a UI framework, such as a command-line tool.
- If you write a loop that creates many temporary objects. You may create an autorelease pool inside the loop to dispose of those objects before the next iteration. Using an autorelease pool in the loop helps to reduce the maximum memory footprint of the application.
- If you spawn a secondary thread.

105- InApp purchase product type

- **Consumable** products must be purchased each time the user needs that item. For example, one-time services are commonly implemented as consumable products.
- **Non-consumable** products are purchased only once by a particular user. Once a non-consumable product is purchased, it is provided to all devices associated with that user's iTunes account. Store Kit provides built-in support to restore non-consumable products on multiple devices.
- **Auto-renewable subscriptions** are delivered to all of a user's devices in the same way as non-consumable products. However, auto-renewable subscriptions differ in other ways. When you create an auto-renewable subscription in iTunes Connect, you choose the duration of the subscription. The App Store automatically renews the subscription each time its term expires. If the user chooses to not allow the subscription to be renewed, the user's access to the subscription is revoked after the subscription expires. Your application is responsible for validating whether a subscription is currently active and can also receive an updated receipt for the most recent transaction.
- **Free subscriptions** are a way for you to put free subscription content in Newsstand. Once a user signs up for a free subscription, the content is available on all devices associated with the user's Apple ID. Free subscriptions do not expire and can only be offered in Newsstand-enabled apps

106-the advantages and disadvantages about synchronous versus asynchronous connections.
That's it, pretty fast and easy, but there are a lot of caveats :

• The most important problem is that the thread which called this method will be blocked until the connection finish or timeout, so we surely don't want to start the connection on the main thread to avoid freezing the UI. That means we need to create a new thread to handle the connection, and all programmers know that threading is hard.

• Cancellation, it's not possible to cancel a synchronous connection, which is bad because users like to have the choice to cancel an operation if they think it takes too much time to execute.

• Authentication, there is no way to deal with authentication challenges.

• It's impossible to parse data on the fly.

So let's put it up straight, avoid using synchronousNSURLConnection, there is absolutely no benefit of using it.
It's clear that **asynchronous connections** give us more control :

• You don't have to create a new thread for the connection because your main thread will not be blocked.

• You can easily cancel the connection just by calling the cancel method.

• If you need authentication just implement the required delegate methods.

• Parsing data on the fly is easy.

So clearly we have a lot of more control with this, and the code is really not difficult.

Even better, we don't have to handle the creation of a new thread, which is a good thing, because you know, threading is hard.

Well, if you read me until here, you should be convinced to use asynchronous connections, and forget about synchronous ones. They clearly give us more control and possibilities and, in some case can spare us to create new thread.

So I encourage you to move away from synchronous connections, just think of

them as evil.

107-What is the navigation controller?

Navigation controller contains the stack of controllers every navigation controller must be having root view controller by default these controllers contain 2 method (a) push view (b) pop view By default navigation controller contain “table view”.

108- What is the split view controller?

This control is used for iPad application and it contain proper controllers by default split view controller contain root view controller and detail view controller.

109-Cocoa.

Cocoa is an application environment for both the Mac OS X operating system and iOS. It consists of a suite of object-oriented software libraries, a runtime system, and an integrated development environment. Carbon is an alternative environment in Mac OS X, but it is a compatibility framework with procedural programmatic interfaces intended to support existing Mac OS X code bases.

110- Frameworks that make Cocoa.

Appkit (Application Kit)
Foundation

111- Objective-C.

Objective-C is a very dynamic language. Its dynamism frees a program from compile-time and link-time constraints and shifts much of the responsibility for symbol resolution to runtime, when the user is in control. Objective-C is more dynamic than other programming languages because its dynamism springs from three sources:
Dynamic typing—determining the class of an object at runtime
Dynamic binding—determining the method to invoke at runtime
Dynamic loading—adding new modules to a program at runtime

112- Objective-C vs C/C++.

· The Objective-C class allows a method and a variable with the exact same name. In C++, they must be different.
· Objective-C does not have a constructor or destructor. Instead it has init and dealloc methods, which must be called explicitly.
· Objective-C uses + and – to differentiate between factory and instance methods, C++ uses static to specify a factory method.
· Multiple inheritance is not allowed in Obj-C, however we can use protocol to some extent.
· Obj-C has runtime binding leading to dynamic linking.
· Obj-C has got categories.
· Objective-C has a work-around for method overloading, but none for operator overloading.
· Objective-C also does not allow stack based objects. Each object must be a pointer to a block of memory.
· In Objective-C the message overloading is faked by naming the parameters. C++ actually does the same thing but the compiler does the name mangling for us. In Objective-C, we have to mangle the names manually.

· One of C++'s advantages and disadvantages is automatic type coercion.
· Another feature C++ has that is missing in Objective-C is references. Because pointers can be used wherever a reference is used, there isn't much need for references in general.
· Templates are another feature that C++ has that Objective-C doesn't. Templates are needed because C++ has strong typing and static binding that prevent generic classes, such as List and Array.

113-Appilcation Kit/App kit.

The Application Kit is a framework containing all the objects you need to implement your graphical, event-driven user interface: windows, panels, buttons, menus, scrollers, and text fields. The Application Kit handles all the details for you as it efficiently draws on the screen, communicates with hardware devices and screen buffers, clears areas of the screen before drawing, and clips views.
You also have the choice at which level you use the Application Kit:
· Use Interface Builder to create connections from user interface objects to your application objects.
· Control the user interface programmatically, which requires more familiarity with AppKit classes and protocols.
· Implement your own objects by subclassing NSView or other classes.

114-Foundation Kit.

The Foundation framework defines a base layer of Objective-C classes. In addition to providing a set of useful primitive object classes, it introduces several paradigms that define functionality not covered by the Objective-C language. The Foundation framework is designed with these goals in mind:
· Provide a small set of basic utility classes.
· Make software development easier by introducing consistent conventions for things such as deallocation.
· Support Unicode strings, object persistence, and object distribution.
· Provide a level of OS independence, to enhance portability.

115-Dynamic and Static Typing.

Static typed languages are those in which type checking is done at compile-time, whereas dynamic typed languages are those in which type checking is done at run-time.
Objective-C is a dynamically-typed language, meaning that you don't have to tell the compiler what type of object you're working with at compile time. Declaring a type for a varible is merely a promise which can be broken at runtime if the code leaves room for such a thing. You can declare your variables as type id, which is suitable for any Objective-C object.

116-Selectors

In Objective-C, selector has two meanings. It can be used to refer simply to the name of a method when it's used in a source-code message to an object. It also, though, refers to the unique identifier that replaces the name when the source code is compiled. Compiled selectors are of type SEL. All methods with the same name have the same selector. You can use a selector to invoke a method on an object—this provides the basis for the implementation of the

target-action design pattern in Cocoa.
[friend performSelector:@selector(gossipAbout:) withObject:aNeighbor];
is equivalent to:
[friend gossipAbout:aNeighbor];

117-Class Introspection

· Determine whether an objective-C object is an instance of a class
[obj isMemberOfClass:someClass];
· Determine whether an objective-C object is an instance of a class or its descendants
[obj isKindOfClass:someClass];
· The version of a class
[MyString version]
· Find the class of an Objective-C object
Class c = [obj1 class]; Class c = [NSString class];
· Verify 2 Objective-C objects are of the same class
[obj1 class] == [obj2 class]

118- Proxy

As long as there aren't any extra instance variables, any subclass can proxy itself as its superclass with a single call. Each class that inherits from the superclass, no matter where it comes from, will now inherit from the proxied subclass. Calling a method in the superclass will actually call the method in the subclass. For libraries where many objects inherit from a base class, proxying the superclass can be all that is needed.

119- Why category is better than inheritance?

If category is used, you can use same class, no need to remember a new class-name. Category created on a base class is available on sub classes.

120-Formal Protocols

Formal Protocols allow us to define the interface for a set of methods, but implementation is not done. Formal Protocols are useful when you are using DistributedObjects, because they allow you to define a protocol for communication between objects, so that the DO system doesn't have to constantly check whether or not a certain method is implemented by the distant object.

121- Formal vs informal protocol.

In addition to formal protocols, you can also define an informal protocol by grouping the methods in a category declaration:
@interface NSObject (MyProtocol)
//someMethod();
@end
Informal protocols are typically declared as categories of the NSObject class, because that broadly associates the method names with any class that inherits from NSObject. Because all classes inherit from the root class, the methods aren't restricted to any part of the inheritance hierarchy. (It is also possible to declare an informal protocol as a category of another class to limit it to a certain branch of the inheritance hierarchy, but there is little reason to do so.)

When used to declare a protocol, a category interface doesn't have a corresponding implementation. Instead, classes that implement the protocol declare the methods again in their own interface files and define them along with other methods in their implementation files.
An informal protocol bends the rules of category declarations to list a group of methods but not associate them with any particular class or implementation. Being informal, protocols declared in categories don't receive much language support. There's no type checking at compile time nor a check at runtime to see whether an object conforms to the protocol. To get these benefits, you must use a formal protocol. An informal protocol may be useful when all the methods are optional, such as for a delegate, but (in Mac OS X v10.5 and later) it is typically better to use a formal protocol with optional methods.

122- Optional vs required

Protocol methods can be marked as optional using the @optional keyword. Corresponding to the @optional modal keyword, there is a @required keyword to formally denote the semantics of the default behaviour. You can use @optional and @required to partition your protocol into sections as you see fit. If you do not specify any keyword, the default is @required.

```
@protocol MyProtocol
@optional
-(void) optionalMethod;
@required
-(void) requiredMethod;
@end
```

123- Memory Management

If you alloc, retain, or copy/mutable copy it, it's your job to release it. Otherwise it isn't.

124-Copy vs assign vs retain

· Assign is for primitive values like BOOL, NSInteger or double. For objects use retain or copy, depending on if you want to keep a reference to the original object or make a copy of it.

· **assign**: In your setter method for the property, there is a simple assignment of your instance variable to the new value, eg:

```
(void)setString:(NSString*)newString{
string = newString;
  }
```

This can cause problems since Objective-C objects use reference counting, and therefore by not retaining the object, there is a chance that the string could be deallocated whilst you are still using it.

· **retain**: this retains the new value in your setter method. For example:
This is safer, since you explicitly state that you want to maintain a reference of the object, and you must release it before it will be deallocated.

```
(void)setString:(NSString*)newString{
[newString retain];
[string release];
```

```
string = newString;
}
```

· **copy**: this makes a copy of the string in your setter method:
This is often used with strings, since making a copy of the original object ensures that it is not changed whilst you are using it.

```
(void)setString:(NSString*)newString{
if(string!=newString){
[string release];
string = [newString copy];
}
   }
```

125- alloc vs new

“alloc” creates a new memory location but doesn’t initializes it as compared to “new”.

126- release vs pool drain

“release” frees a memory. “drain” releases the NSAutoreleasePool itself.

127- NSAutoReleasePool : release vs drain

Strictly speaking, from the big picture perspective drain is not equivalent to release:
In a reference-counted environment, drain does perform the same operations as release, so the two are in that sense equivalent. To emphasise, this means you do not leak a pool if you use drain rather than release.
In a garbage-collected environment, release is a no-op. Thus it has no effect. drain, on the other hand, contains a hint to the collector that it should “collect if needed”. Thus in a garbage-collected environment, using drain helps the system balance collection sweeps.

128-autorelease vs release

Autorelase: By sending an object an autorelease message, it is added to the local AutoReleasePool, and you no longer have to worry about it, because when the AutoReleasePool is destroyed (as happens in the course of event processing by the system) the object will receive a release message, its RetainCount will be decremented, and the GarbageCollection system will destroy the object if the RetainCount is zero.
Release: retain count is decremented at this point.

129- Autorelease Pool

Autorelease pools provide a mechanism whereby you can send an object a “deferred” release message. This is useful in situations where you want to relinquish ownership of an object, but want to avoid the possibility of it being deallocated immediately (such as when you return an object from a method). Typically, you don’t need to create your own autorelease pools, but there are some situations in which either you must or it is beneficial to do so.

130- How autorelease pool is managed.

Every time -autorelease is sent to an object, it is added to the inner-most autorelease pool. When the pool is drained, it simply sends -release to all the objects in the pool.

Autorelease pools are simply a convenience that allows you to defer sending -release until “later”. That “later” can happen in several places, but the most common in Cocoa GUI apps is at the end of the current run loop cycle.

131-Memory Leak

If RetainingAndReleasing are not properly used then RetainCount for AnObject doesn’t reach 0. It doesn’t crash the application.

132- Event Loop

In a Cocoa application, user activities result in events. These might be mouse clicks or drags, typing on the keyboard, choosing a menu item, and so on. Other events can be generated automatically, for example a timer firing periodically, or something coming in over the network. For each event, Cocoa expects there to be an object or group of objects ready to handle that event appropriately. The event loop is where such events are detected and routed off to the appropriate place. Whenever Cocoa is not doing anything else, it is sitting in the event loop waiting for an event to arrive. (In fact, Cocoa doesn’t poll for events as suggested, but instead its main thread goes to sleep. When an event arrives, the OS wakes up the thread and event processing resumes. This is much more efficient than polling and allows other applications to run more smoothly).

Each event is handled as an individual thing, then the event loop gets the next event, and so on. If an event causes an update to be required, this is checked at the end of the event and if needed, and window refreshes are carried out.

133-difference between boxName and self.boxName.

boxName: Accessing directly.

self. boxName: Accessing boxName through accessors. If property/synthesize is not there it will throw error.

134-What it does “@synthesize boxDescription=boxName;” ?

Here you can use boxName or self.boxName. We cant use boxDescription.

135-Collection

In Cocoa and Cocoa Touch, a collection is a Foundation framework class used for storing and managing groups of objects. Its primary role is to store objects in the form of either an array, a dictionary, or a set.

136-Threads and how to use

Use this class when you want to have an Objective-C method run in its own thread of execution. Threads are especially useful when you need to perform a lengthy task, but don’t want it to block the execution of the rest of the application. In particular, you can use threads to avoid blocking the main thread of the application, which handles user interface and event-related actions. Threads can also be used to divide a large job into several smaller jobs, which can lead to performance increases on multi-core computers.

Two ways to create threads…

· detachNewThreadSelector:toTarget:withObject:

· Create instances of NSThread and start them at a later time using the “start” method.

NSThread is not as capable as Java’s Thread class, it lacks

· Built-in communication system.

· An equivalent of “join()”

137-Threadsafe

When it comes to threaded applications, nothing causes more fear or confusion than the issue of handling signals. Signals are a low-level BSD mechanism that can be used to deliver information to a process or manipulate it in some way. Some programs use signals to detect certain events, such as the death of a child process. The system uses signals to terminate runaway processes and communicate other types of information.

The problem with signals is not what they do, but their behaviour when your application has multiple threads. In a single-threaded application, all signal handlers run on the main thread. In a multithreaded application, signals that are not tied to a specific hardware error (such as an illegal instruction) are delivered to whichever thread happens to be running at the time. If multiple threads are running simultaneously, the signal is delivered to whichever one the system happens to pick. In other words, signals can be delivered to any thread of your application.

The first rule for implementing signal handlers in applications is to avoid assumptions about which thread is handling the signal. If a specific thread wants to handle a given signal, you need to work out some way of notifying that thread when the signal arrives. You cannot just assume that installation of a signal handler from that thread will result in the signal being delivered to the same thread.

138-Notification and Observers

A notification is a message sent to one or more observing objects to inform them of an event in a program. The notification mechanism of Cocoa follows a **broadcast** model. It is a way for an object that initiates or handles a program event to communicate with any number of objects that want to know about that event. These recipients of the notification, known as observers, can adjust their own appearance, behaviour, and state in response to the event. The object sending (or posting) the notification doesn’t have to know what those observers are. Notification is thus a powerful mechanism for attaining coordination and cohesion in a program. It reduces the need for strong dependencies between objects in a program (such dependencies would reduce the reusability of those objects). Many classes of the Foundation, AppKit, and other Objective-C frameworks define notifications that your program can register to observe.

The centre piece of the notification mechanism is a per-process singleton object known as the notification centre (**NSNotificationCenter**). When an object posts a notification, it goes to the notification centre, which acts as a kind of clearing house and broadcast centre for notifications. Objects that need to know about an event elsewhere in the application register with the notification centre to let it know they want to be notified when that event happens. Although the notification centre delivers a notification to its observers synchronously, you can post notifications asynchronously using a notification queue (NSNotificationQueue).

139-Delegate vs Notification

· The concept of notification differs from delegation in that it allows a message to be sent to more than one object. It is more like a broadcast rather than a straight communication between two objects. It removes dependencies between the sending and receiving object(s) by using a notification centre to manage the sending and receiving of notifications. The sender does not need to know if there are any receivers registered with the notification centre. There can be one, many or even no receivers of the notification registered with the notification centre. Simply, Delegate is 1-to-1 object and Notification can be *-to-* objects.
· The other difference between notifications and delegates is that there is no possibility for the receiver of a notification to return a value to the sender.
· Typical uses of notifications might be to allow different objects with an application to be informed of an event such as a file download completing or a user changing an application preference. The receiver of the notification might then perform additional actions such as processing the downloaded file or updating the display.

140-Plist

Property lists organise data into named values and lists of values using several object types. These types give you the means to produce data that is meaningfully structured, transportable, storable, and accessible, but still as efficient as possible. Property lists are frequently used by applications running on both Mac OS X and iOS. The property-list programming interfaces for Cocoa and Core Foundation allow you to convert hierarchically structured combinations of these basic types of objects to and from standard XML. You can save the XML data to disk and later use it to reconstruct the original objects.
The user defaults system, which you programmatically access through the NSUserDefaults class, uses property lists to store objects representing user preferences. This limitation would seem to exclude many kinds of objects, such as NSColor and NSFont objects, from the user default system. But if objects conform to the NSCoding protocol they can be archived to NSData objects, which are property list–compatible objects

141-Helper Objects

Helper Objects are used throughout Cocoa and CocoaTouch, and usually take the form of a delegate or dataSource. They are commonly used to add functionality to an existing class without having to subclass it.

142-Cluster Class

Class clusters are a design pattern that the Foundation framework makes extensive use of. Class clusters group a number of private concrete subclasses under a public abstract superclass. The grouping of classes in this way simplifies the publicly visible architecture of an object-oriented framework without reducing its functional richness.

143-Differentiate Foundation vs Core Foundation

CoreFoundation is a general-purpose C framework whereas Foundation is a general-purpose Objective-C framework. Both provide collection classes, run loops, etc, and many of the Foundation classes are wrappers around the CF

equivalents. CF is mostly open-source , and Foundation is closed-source. **Core Foundation** is the C-level API, which provides CFString, CFDictionary and the like.**Foundation** is Objective-C, which provides NSString, NSDictionary, etc. CoreFoundation is written in C while Foundation is written in Objective-C. Foundation has a lot more classes CoreFoundation is the common base of Foundation and Carbon.

144-Difference between coreData and Database

Database	Core Data
Primary function is storing and fetching data	Primary function is graph management (although reading and writing to disk is an important supporting feature)
Operates on data stored on disk (or minimally and incrementally loaded)	Operates on objects stored in memory (although they can be lazily loaded from disk)
Stores “dumb” data	Works with fully-fledged objects that self-manage a lot of their behaviour and can be subclassed and customised for further behaviours
Can be transactional, thread-safe, multi-user	Non-transactional, single threaded, single user (unless you create an entire abstraction around Core Data which provides these things)
Can drop tables and edit data without loading into memory	Only operates in memory
Perpetually saved to disk (and often crash resilient)	Requires a save process
Can be slow to create millions of new rows	Can create millions of new objects in-memory very quickly (although saving these objects will be slow)
Offers data constraints like “unique” keys	Leaves data constraints to the business logic side of the program

145- Core data vs sqlite.

Core data is an object graph management framework. It manages a potentially very large graph of object instances, allowing an app to work with a graph that would not entirely fit into memory by faulting objects in and out of memory as necessary. Core Data also manages constraints on properties and relationships and maintains reference integrity (e.g. keeping forward and backwards links consistent when objects are added/removed to/from a relationship). Core Data is thus an ideal framework for building the “model” component of an MVC architecture.

To implement its graph management, Core Data happens to use sqlite as a disk store. It could have been implemented using a different relational database or even a non-relational database such as CouchDB. As others have pointed out, Core Data can also use XML or a binary format or a

user-written atomic format as a backend (though these options require that the entire object graph fit into memory).

146-Retain cycle or Retain loop.

When object A retains object B, and object B retains A. Then Retain cycle happens. To overcome this use “close” method.

Objective-C’s garbage collector (when enabled) can also delete retain-loop groups but this is not relevant on the iPhone, where Objective-C garbage collection is not supported.

147-What is unnamed category.

A named category — **@interface Foo(FooCategory)** — is generally used to:

i. Extend an existing class by adding functionality.

ii. Declare a set of methods that might or might not be implemented by a delegate.

Unnamed Categories has fallen out of favour now that @protocol has been extended to support @optional methods.

A class extension — **@interface Foo()** — is designed to allow you to declare additional private API — SPI or System Programming Interface — that is used to implement the class innards. This typically appears at the top of the .m file. Any methods / properties declared in the class extension must be implemented in the @implementation, just like the methods/properties found in the public @interface.

Class extensions can also be used to redeclare a publicly readonly @property as readwrite prior to @synthesize’ing the accessors.

Example:

Foo.h

```
@interface Foo:NSObject
@property(readonly, copy) NSString *bar;
-(void) publicSaucing;
@end
```

Foo.m

```
@interface Foo()
@property(readwrite, copy) NSString *bar;
– (void) superSecretInternalSaucing;
@end
@implementation Foo
@synthesize bar;
…. must implement the two methods or compiler will warn ….
@end
```

148-Copy vs mutableCopy.

copy always creates an immutable copy.

mutableCopy always creates a mutable copy.

149- Strong vs Weak

The strong and weak are new ARC types replacing retain and assign respectively.

Delegates and outlets should be weak.

A **strong reference** is a reference to an object that stops it from being

deallocated. In other words it creates a owner relationship.
A **weak reference** is a reference to an object that does not stop it from being deallocated. In other words, it does not create an owner relationship.
150-__strong, __weak, __unsafe_unretained, __autoreleasing.
Generally speaking, these extra qualifiers don't need to be used very often. You might first encounter these qualifiers and others when using the migration tool. For new projects however, you generally you won't need them and will mostly use strong/weak with your declared properties.
__strong – is the default so you don't need to type it. This means any object created using alloc/init is retained for the lifetime of its current scope. The "current scope" usually means the braces in which the variable is declared
__weak – means the object can be destroyed at anytime. This is only useful if the object is somehow strongly referenced somewhere else. When destroyed, a variable with __weak is set to nil.
__unsafe_unretained – is just like __weak but the pointer is not set to nil when the object is deallocated. Instead the pointer is left dangling.
__autoreleasing, not to be confused with calling autorelease on an object before returning it from a method, this is used for passing objects by reference, for example when passing NSError objects by reference such as [myObject performOperationWithError:&tmp];
151-Types of NSTableView
Cell based and View based. In view based we can put multiple objects.
152-Abstract class in cocoa.
Cocoa doesn't provide anything called abstract. We can create a class abstract which gets check only at runtime, compile time this is not checked.

```
@interface AbstractClass : NSObject
@end
@implementation AbstractClass
+ (id)alloc{
if (self == [AbstractClass class]) {
NSLog(@"Abstract Class cant be used");
}
return [super alloc];
@end
```

153- Difference between HTTP and HTTPS.
· HTTP stands for HyperText Transfer Protocol, whereas, HTTPS is HyperText Transfer Protocol Secure.
· HTTP transmits everything as plan text, while HTTPS provides encrypted communication, so that only the recipient can decrypt and read the information. Basically, HTTPS is a combination of HTTP and SSL (Secure Sockets Layer). This SSL is that protocol which encrypts the data.
· HTTP is fast and cheap, where HTTPS is slow and expensive.
As, HTTPS is safe it's widely used during payment transactions or any sensitive transactions over the internet. On the other hand, HTTP is used most of the sites over the net, even this blogspot sites also use HTTP.
· HTTP URLs starts with "http:// " and use port 80 by default, while HTTPS

URLs stars with “https:// “ and use port 443.
· HTTP is unsafe from attacks like man-in-the-middle and eavesdropping, but HTTPS is secure from these sorts of attacks.

154-GCD

Grand Central Dispatch is not just a new abstraction around what we’ve already been using, it’s an entire new underlying mechanism that makes multithreading easier and makes it easy to be as concurrent as your code can be without worrying about the variables like how much work your CPU cores are doing, how many CPU cores you have and how much threads you should spawn in response. You just use the Grand Central Dispatch API’s and it handles the work of doing the appropriate amount of work. This is also not just in Cocoa, anything running on Mac OS X 10.6 Snow Leopard can take advantage of Grand Central Dispatch (libdispatch) because it’s included in libSystem.dylib and all you need to do is include #import <dispatch/dispatch.h> in your app and you’ll be able to take advantage of Grand Central Dispatch.

155-How you attain the backward compatibility?

- Set the Base SDK to Current version of Mac (ex. 10.7)
- Set the Deployment SDK to older version (ex.1.4)

156-Call Back.

Synchronous operations are ones that happen in step with your calling code. Most of Cocoa works this way: you send a message to an object, say to format a string, etc, and by the time that line of code is “done”, the operation is complete.
But in the real world, some operations take longer than “instantaneous” (some intensive graphics work, but mainly high or variably latency things like disk I/O or worse, network connectivity). These operations are unpredictable, and if the code were to block until finish, it might block indefinitely or forever, and that’s no good.
So the way we handle this is to set up “callbacks”– you say “go off and do this operation, and when you’re done, call this other function”. Then inside that “callback” function, you start the second operation that depends on the first. In this way, you’re not spinning in circles waiting, you just get called “asynchronously” when each task is done.

IOS INTERVIEW QUESTIONS WITH ANSWERS

1.What is latest iOS version?

IOS – 7.1

2.What is latest Xcode version?

Xcode- 5

3.What is latest mac os version?

Mavericks

4.What is iPad screen size?
1024X768

5.what is iPhone screen size?
320X480

6.What are the features is IOS 6?
1.Map :beautifully designed from the ground up (and the sky down)
2.Integration of Facebook with iOS
3.shared photo streams.
4.Passbook - boarding passes, loyalty cards, retail coupons, cinema tickets and more all
in one place
5.Facetime - on mobile network as wifi
6.changed Phone app - *remind me later,*reply with message.
7.Mail - redesigned more streamline interface.
8.Camera with panorama .

7.Who invented Objective c?
Broad cox and Tom Love

8.What is Cococa and cocoa touch?
Cocoa is for Mac App development and cocoa touch is for apples touch devices - that
provide all development environment

9.What is Objective c?
*Objective-C is a reflective, object-oriented programming language which adds
Smalltalk-style messaging to the C programming language. strictly superset of c.

10. how declare methods in Objective c? and how to call them?
- (return_type)methodName:(data_type)parameter_name :(data_type)parameter_name

11. What is property in Objective c?
Property allow declared variables with specification like atomic/nonatmic, or retain/assign

12.What is meaning of "copy" keyword?
copy object during assignment and increases retain count by 1

13.What is meaning of "readOnly" keyword?
Declare read only object / declare only getter method

14.What is meaning of "retain" keyword?
Specifies that retain should be invoked on the object upon assignment. takes ownership
of an object

15.What is meaning of "assign" keyword?
Specifies that the setter uses simple assignment. Uses on attribute of scalar type like
float,int.
Answers:
iOS interview questions with answers http://abhijeetbargeios.blogspot.in/
2 of 10 26/11/13 11:46 AM

16.What is meaning of "atomic" keyword?
"atomic", the synthesized setter/getter will ensure that a whole value is always returned
from the getter or set by the setter, only single thread can access variable to get or set
value at a time

17.What is meaning of "nonatomic" keyword?
In non atomic no such guaranty that value is returned from variable is same that setter
sets. at same time

18.What is difference between "assign" and "retain" keyword?
Retain -Specifies that retain should be invoked on the object upon assignment. takes
ownership of an object
Assign - Specifies that the setter uses simple assignment. Uses on attribute of scalar type
like float,int.

19.What is meaning of "synthesize" keyword ?
ask the compiler to generate the setter and getter methods according to the specification
in the declaration

20.What is "Protocol" on objective c?
A protocol declares methods that can be implemented by any class. Protocols are not
classes themselves. They simply define an interface that other objects are responsible for

implementing. Protocols have many advantages. The idea is to provide a way for classes
to share the same method and property declarations without inheriting them from a
common ancestor

21.What is use of UIApplication class?
The UIApplication class implements the required behavior of an application.

22.What compilers apple using ?
The Apple compilers are based on the compilers of the GNU Compiler Collection.

23.What is synchronized() block in objective c? what is the use of that?
The @synchronized()directive locks a section of code for use by a single thread. Other
threads are blocked until the thread exits the protected code

.
24. What is the "interface" and "implementation"?
interface declares the behavior of class and implementation defines the behavior of
class.

25.What is "private", "Protected" and "Public" ?
private - limits the scope class variable to the class that declares it.
protected - Limits instance variable scope to declaring and inheriting classes.
public - Removes restrictions on the scope of instance variables

26. What is the use of "dynamic" keyword?
Instructs the compiler not to generate a warning if it cannot find implementations of
accessor methods associated with the properties whose names follow.

27.What is "Delegate" ?
A delegate is an object that will respond to pre-chosen selectors (function calls) at some
point in the future., need to implement the protocol method by the delegate object.

28.What is "notification"?
provides a mechanism for broadcasting information within a program, using notification
we can send message to other object by adding observer .

29.What is difference between "protocol" and "delegate"?

protocol is used the declare a set of methods that a class that "adopts" (declares that it
will use this protocol) will implement.
Delegates are a use of the language feature of protocols. The delegation design pattern is
a way of designing your code to use protocols where necessary.

30.What is "Push Notification"?
to get the any update /alert from server .

31.How to deal with SQLite database?
Dealing with sqlite database in iOS:
1. Create database : sqlite3 AnimalDatabase.sql
2.Create table and insert data in to table :
CREATE TABLE animals (id INTEGER PRIMARY KEY, name VARCHAR(50),
description TEXT, image VARCHAR(255));
iOS interview questions with answers http://abhijeetbargeios.blogspot.in/
3 of 10 26/11/13 11:46 AM
INSERT INTO animals (name, description, image) VALUES ('Elephant', 'The elephant
is a very large animal that lives in Africa and Asia',
'http://dblog.com.au/wp-content
/elephant.jpg');
3. Create new app --> Add SQLite framework and database file to project
4. Read the database and close it once work done with database :

```
// Setup the database object
sqlite3 *database;
// Init the animals Array
animals = [[NSMutableArray alloc] init];
// Open the database from the users filessytem
if(sqlite3_open([databasePath UTF8String], &database) == SQLITE_OK) {
// Setup the SQL Statement and compile it for faster access
const char *sqlStatement = "select * from animals";
sqlite3_stmt *compiledStatement;
if(sqlite3_prepare_v2(database, sqlStatement, -1, &compiledStatement, NULL) ==
SQLITE_OK) {
// Loop through the results and add them to the feeds array
while(sqlite3_step(compiledStatement) == SQLITE_ROW) {
// Read the data from the result row
NSString *aName = [NSString stringWithUTF8String:(char
*)sqlite3_column_text(compiledStatement, 1)];
NSString *aDescription = [NSString stringWithUTF8String:(char
*)sqlite3_column_text(compiledStatement, 2)];
NSString *aImageUrl = [NSString stringWithUTF8String:(char
```

```
*)sqlite3_column_text(compiledStatement, 3)];
// Create a new animal object with the data from the database
Animal *animal = [[Animal alloc] initWithName:aName
description:aDescription
url:aImageUrl];
// Add the animal object to the animals Array
[animals addObject:animal];
[animal release];
}
}
// Release the compiled statement from memory
sqlite3_finalize(compiledStatement);
}
sqlite3_close(database);
```

32.What is storyboard?

With Storyboards, all screens are stored in a single file. This gives you a conceptual overview of the visual representation for the app and shows you how the screens are connected. Xcode provides a built-in editor to layout the Storyboards. .storyboard is essentially one single file for all your screens in the app and it shows the flow of the screens. You can add segues/transitions between screens, this way. So, this minimizes the boilerplate code required to manage multiple screens.

1.

2. Minimizes 2. the overall no. of files in an app.

33.What is Category in Objective c?

A category allows you to add methods to an existing class—even to one for which you do not have the source.

34.What is block in objective c?

Blocks are a language-level feature added to C, Objective-C and C++, which allow you to create distinct segments of code that can be passed around to methods or functions as if they were values. Blocks are Objective-C objects, which means they can be added to collections like NSArray or NSDictionary. They also have the ability to capture values from the enclosing scope, making them similar to closures or lambdas in other programming languages.

35. How to parse xml? explain in deep.
Using NSXMLParser.
Create xml parser object with xml data, set its delegate , and call the parse method with
iOS interview questions with answers http://abhijeetbargeios.blogspot.in/
4 of 10 26/11/13 11:46 AM
parserObject.
Delegate methods getting called :
– parserDidStartDocument:
– parserDidEndDocument:
– parser:didStartElement:namespaceURI:qualifiedName:attributes:
– parser:didEndElement:namespaceURI:qualifiedName:
– parser:didStartMappingPrefix:toURI:
– parser:didEndMappingPrefix:
– parser:resolveExternalEntityName:systemID:
– parser:parseErrorOccurred:
– parser:validationErrorOccurred:
– parser:foundCharacters:
– parser:foundIgnorableWhitespace:
– parser:foundProcessingInstructionWithTarget:data:
– parser:foundComment:
– parser:foundCDATA:

36.How to parse JSON? explain in deep.
By using NSJSONSerialization.
For example : NSArray *jsonArray = [NSJSONSerialization JSONObjectWithData:
data options: NSJSONReadingMutableContainers error: &e];

37.How to use reusable cell in UITableview?
By using dequeReusableCellWithIdentifier

38.What is the meaning of "strong"keyword?
*strong -o "own" the object you are referencing with this property/variable. The compiler will take care that any object that you assign to this property will not be
destroyed as long as you (or any other object) points to it with a strong reference.

39.What is the meaning of "weak" keyword?
*Weak - weak reference you signify that you don't want to have control over the object's
lifetime. The object you are referencing weakly only lives on because at least

one other
object holds a strong reference to it. Once that is no longer the case, the object gets
destroyed and your weak property will automatically get set to nil.

40.What is difference strong and weak reference ? explain.
complier with be responsible for lifetime of object which is declared as strong. for weak
object - compiler will destroy object once strong reference that hold weak object get
destroyed.

41.What is ARC ? How it works? explain in deep.
Automatic reference counting (ARC) If the compiler can recognize where you should
be retaining and releasing objects, and put the retain and release statement in code.

42. What manual memory management ? how it work?
In Manual memory management developers is responsible for life cycle of object.
developer has to retain /alloc and release the object wherever needed.

43. How to find the memory leaks in MRC?
By using -
1. Static analyzer.
2. Instrument

44.what is use of NSOperation? how NSOperationque works?
An operation object is a single-shot object—that is, it executes its task once and cannot
be used to execute it again. You typically execute operations by adding them to an
operation queueAn NSOperationQueue object is a queue that handles objects of the
NSOperation class type. An NSOperation object, simply phrased, represents a single
task, including both the data and the code related to the task. The NSOperationQueue
handles and manages the execution of all the NSOperation objects (the tasks) that have
been added to it.

45.How to send crash report from device?

46.What is autorealease pool?
Every time -autorelease is sent to an object, it is added to the inner-most autorelease
pool. When the pool is drained, it simply sends -release to all the objects in the pool.
Autorelease pools are simply a convenience that allows you to defer sending -release
until "later". That "later" can happen in several places, but the most common in Cocoa
GUI apps is at the end of the current run loop cycle.

47.What happens when we invoke a method on a nil pointer?

48.Difference between nil and Nil.
Nil is meant for class pointers, and nil is meant for object pointers
iOS interview questions with answers http://abhijeetbargeios.blogspot.in/
5 of 10 26/11/13 11:46 AM

49.What is fast enumeration?
for(id object in objets){
}

50. How to start a thread?
- (void)performSelectorInBackground:(SEL)aSelector withObject:(id)arg on NSObject
NSThread* evtThread = [[NSThread alloc] initWithTarget:self
selector:@selector(saySomething)
object:nil];
[evtThread start];

51.How to download something from the internet?
By Using NSURLConnection , by starting connection or sending synchronous request.

52.what is synchronous web request and asynchronous ?
In synchronous request main thread gets block and control will not get back to user till
that request gets execute.
In Asynchronous control gets back to user even if request is getting execute.

53. Difference between sax parser and dom parser ?
SAX (Simple API for XML)

1. Parses node by node
2. Doesn't store the XML in memory
3. We can not insert or delete a node
4. Top to bottom traversing

DOM (Document Object Model)

1. Stores the entire XML document into memory before processing
2. Occupies more memory
3. We can insert or delete nodes
4. Traverse in any direction

54.Explain stack and heap?

55.What are the ViewController lifecycle in iOS?

loadView - viewDidLoad-viewWillAppear-viewDidAppear - viewDisappear - viewDidUnload

56.Difference between core data & sqlite?

There is a huge difference between these two. SQLite is a database itself like we have
MS SQL Server. But CoreData is an ORM (Object Relational Model) which creates a
layer between the database and the UI. It speeds-up the process of interaction as we don’t
have to write queries, just work with the ORM and let ORM handles the backend. For
save or retrieval of large data, I recommend to use Core Data because of its abilities to
handle the less processing speed of iPhone.

57.Steps for using core data?

NSFetchedResultsController - It is designed primarily to function as a data source for a
UITableView

58.Procedure to push the app in AppStore?

59.What are the Application lifecycle in ios?

ApplicationDidFinishLaunchingWithOption -ApplicationWillResignActive-ApplicationDidBecomeActive-ApplicationWillTerminate

60.Difference between release and autorelease ?

release - destroy the object from memory,
autorelease - destroy the object from memory in future when it is not in use.

61.How to start a selector on a background thread

- (void)performSelectorInBackground:(SEL)aSelector withObject:(id)arg on

NSObject

62.What happens if the methods doesn't exist
iOS interview questions with answers http://abhijeetbargeios.blogspot.in/
6 of 10 26/11/13 11:46 AM
App will crash with exception unrecognized selector sent to instance.

63. How Push notification works?
Server - Apple server - device by using APNs

Delegate methods :
UITableView:
DataSource -
Configuring a Table View
– tableView:cellForRowAtIndexPath: required method
– numberOfSectionsInTableView:
– tableView:numberOfRowsInSection: required method
– sectionIndexTitlesForTableView:
– tableView:sectionForSectionIndexTitle:atIndex:
– tableView:titleForHeaderInSection:
– tableView:titleForFooterInSection:
Inserting or Deleting Table Rows
– tableView:commitEditingStyle:forRowAtIndexPath:
– tableView:canEditRowAtIndexPath:
Reordering Table Rows
– tableView:canMoveRowAtIndexPath:
– tableView:moveRowAtIndexPath:toIndexPath:
Delegate -
Configuring Rows for the Table View
– tableView:heightForRowAtIndexPath:
– tableView:indentationLevelForRowAtIndexPath:
– tableView:willDisplayCell:forRowAtIndexPath:
Managing Accessory Views
– tableView:accessoryButtonTappedForRowWithIndexPath:
Managing Selections
– tableView:willSelectRowAtIndexPath:
– tableView:didSelectRowAtIndexPath:
– tableView:willDeselectRowAtIndexPath:
– tableView:didDeselectRowAtIndexPath:
Modifying the Header and Footer of Sections
– tableView:viewForHeaderInSection:
– tableView:viewForFooterInSection:
– tableView:heightForHeaderInSection:

– tableView:heightForFooterInSection:
Editing Table Rows
– tableView:willBeginEditingRowAtIndexPath:
– tableView:didEndEditingRowAtIndexPath:
– tableView:editingStyleForRowAtIndexPath:
– tableView:titleForDeleteConfirmationButtonForRowAtIndexPath:
– tableView:shouldIndentWhileEditingRowAtIndexPath:
Reordering Table Rows
– tableView:targetIndexPathForMoveFromRowAtIndexPath:toPropo
sedIndexPath:
Copying and Pasting Row Content
– tableView:shouldShowMenuForRowAtIndexPath:
– tableView:canPerformAction:forRowAtIndexPath:withSender:
– tableView:performAction:forRowAtIndexPath:withSender:

```
- (void)scrollViewDidScroll:(UIScrollView *)scrollView;
// any offset changes
- (void)scrollViewDidZoom:(UIScrollView *)scrollView
NS_AVAILABLE_IOS(3_2); // any zoom scale changes

// called on start of dragging (may require some time and or distance to
move)
- (void)scrollViewWillBeginDragging:(UIScrollView *)scrollView;
// called on finger up if the user dragged. velocity is in points/millisecond.
targetContentOffset may be changed to adjust where the scroll view
comes to rest
- (void)scrollViewWillEndDragging:(UIScrollView *)scrollView
withVelocity:(CGPoint)velocity targetContentOffset:(inout CGPoint
*)targetContentOffset NS_AVAILABLE_IOS(5_0);
// called on finger up if the user dragged. decelerate is true if it will
continue moving afterwards
- (void)scrollViewDidEndDragging:(UIScrollView *)scrollView
willDecelerate:(BOOL)decelerate;

- (void)scrollViewWillBeginDecelerating:(UIScrollView *)scrollView;    //
called on finger up as we are moving
- (void)scrollViewDidEndDecelerating:(UIScrollView *)scrollView;       //
called when scroll view grinds to a halt

- (void)scrollViewDidEndScrollingAnimation:(UIScrollView *)scrollView; //
called when setContentOffset/scrollRectVisible:animated: finishes. not
called if not animating

- (UIView *)viewForZoomingInScrollView:(UIScrollView *)scrollView;
// return a view that will be scaled. if delegate returns nil, nothing
happens
- (void)scrollViewWillBeginZooming:(UIScrollView *)scrollView
```

withView:(UIView *)view NS_AVAILABLE_IOS(3_2); // called before the scroll view begins zooming its content
- (void)scrollViewDidEndZooming:(UIScrollView *)scrollView withView:(UIView *)view atScale:(CGFloat)scale; // scale between minimum and maximum. called after any 'bounce' animations

- (BOOL)scrollViewShouldScrollToTop:(UIScrollView *)scrollView; // return a yes if you want to scroll to the top. if not defined, assumes YES
- (void)scrollViewDidScrollToTop:(UIScrollView *)scrollView; // called when scrolling animation finished. may be called immediately if already at top

UIPickerView-
DataSource -
Providing Counts for the Picker View
– numberOfComponentsInPickerView:
– pickerView:numberOfRowsInComponent:
Delegate -
Setting the Dimensions of the Picker View
– pickerView:rowHeightForComponent:
– pickerView:widthForComponent:
Setting the Content of Component Rows
The methods in this group are marked @optional. However, to use a picker view, you
must implement either thepickerView:titleForRow:forComponent: or the pickerView:viewForRow:forComponent:reusingView: method to provide the content of
component rows.
– pickerView:titleForRow:forComponent:
– pickerView:viewForRow:forComponent:reusingView:
Responding to Row Selection
– pickerView:didSelectRow:inComponent:
UITextFeild-
Delegate -
iOS interview questions with answers http://abhijeetbargeios.blogspot.in/
7 of 10 26/11/13 11:46 AM
Managing Editing
– textFieldShouldBeginEditing:
– textFieldDidBeginEditing:
– textFieldShouldEndEditing:
– textFieldDidEndEditing:
Editing the Text Field’s Text
– textField:shouldChangeCharactersInRange:replacementString:
– textFieldShouldClear:
– textFieldShouldReturn:
UItextView-

Delegate - Responding to Editing Notifications
– textViewShouldBeginEditing:
– textViewDidBeginEditing:
– textViewShouldEndEditing:
– textViewDidEndEditing:
Responding to Text Changes
– textView:shouldChangeTextInRange:replacementText:
– textViewDidChange:
Responding to Selection Changes
– textViewDidChangeSelection:

MKMapView-

Delegate -
Responding to Map Position Changes
– mapView:regionWillChangeAnimated:
– mapView:regionDidChangeAnimated:
Loading the Map Data
– mapViewWillStartLoadingMap:
– mapViewDidFinishLoadingMap:
– mapViewDidFailLoadingMap:withError:
Tracking the User Location
– mapViewWillStartLocatingUser:
– mapViewDidStopLocatingUser:
– mapView:didUpdateUserLocation:
– mapView:didFailToLocateUserWithError:
– mapView:didChangeUserTrackingMode:animated: required
method
Managing Annotation Views
– mapView:viewForAnnotation:
– mapView:didAddAnnotationViews:
– mapView:annotationView:calloutAccessoryControlTapped:
Dragging an Annotation View
– mapView:annotationView:didChangeDragState:fromOldState:
Selecting Annotation Views
– mapView:didSelectAnnotationView:
– mapView:didDeselectAnnotationView:
Managing Overlay Views
– mapView:viewForOverlay:
– mapView:didAddOverlayViews:

NSURLConnection-

Delegate -
Connection Authentication
– connection:willSendRequestForAuthenticationChallenge:
– connection:canAuthenticateAgainstProtectionSpace:
– connection:didCancelAuthenticationChallenge:
– connection:didReceiveAuthenticationChallenge:
– connectionShouldUseCredentialStorage:

Connection Completion
– connection:didFailWithError:
NSURLConnectionDownloadDelegate
– connection:didWriteData:totalBytesWritten:expectedTotalBytes:
– connectionDidResumeDownloading:totalBytesWritten:expectedTot
alBytes:
– connectionDidFinishDownloading:destinationURL:
NSURLConnection
Preflighting a Request
+ canHandleRequest:
Loading Data Synchronously
+ sendSynchronousRequest:returningResponse:error:
Loading Data Asynchronously

+ connectionWithRequest:delegate:
– initWithRequest:delegate:
– initWithRequest:delegate:startImmediately:
+ sendAsynchronousRequest:queue:completionHandler:
– start
Stopping a Connection
– cancel
Scheduling Delegate Messages
– scheduleInRunLoop:forMode:
– setDelegateQueue:
– unscheduleFromRunLoop:forMode:
NSXMLParser-
Handling XML
– parserDidStartDocument:
– parserDidEndDocument:
– parser:didStartElement:namespaceURI:qualifiedName:attributes:
– parser:didEndElement:namespaceURI:qualifiedName:
– parser:didStartMappingPrefix:toURI:
– parser:didEndMappingPrefix:
– parser:resolveExternalEntityName:systemID:
– parser:parseErrorOccurred:
– parser:validationErrorOccurred:
– parser:foundCharacters:
– parser:foundIgnorableWhitespace:
– parser:foundProcessingInstructionWithTarget:data:
– parser:foundComment:
– parser:foundCDATA:
Handling the DTD
– parser:foundAttributeDeclarationWithName:forElement:type:defaultValue:
– parser:foundElementDeclarationWithName:model:
– parser:foundExternalEntityDeclarationWithName:publicID:systemID:
– parser:foundInternalEntityDeclarationWithName:value:

–
parser:foundUnparsedEntityDeclarationWithName:publicID:systemID:notation
Name:
– parser:foundNotationDeclarationWithName:publicID:systemID:

7.NSURLConnection

Connection Authentication
– connection:willSendRequestForAuthenticationChallenge:
– connection:canAuthenticateAgainstProtectionSpace:
– connection:didCancelAuthenticationChallenge:
– connection:didReceiveAuthenticationChallenge:
– connectionShouldUseCredentialStorage:
Connection Completion
– connection:didFailWithError:
MethodGroup
– connection:needNewBodyStream
– connection:didSendBodyData:totalBytesWritten:totalBytesExpectedToWrite:
required method
– connection:didReceiveData: required method
– connection:didReceiveResponse: required method
– connection:willCacheResponse: required method
– connection:willSendRequest:redirectResponse: required method
– connectionDidFinishLoading: required method

https://springpad.com/#!/jonathon.reedy/explore/iosdevelopment/blocks/book
mark/iosinterviewquestionswithanswers