Variable property attributes or Modifiers in iOS

Variable property attributes or Modifiers

Property Attributes Indicate Data Accessibility and Storage Considerations

Use Accessor Methods to Get or Set Property Values

01. atomic //default
02. nonatomic
03. strong=retain //default
04. weak= unsafe_unretained
05. retain
06. assign //default
07. unsafe_unretained
08. copy
09. readonly
10. readwrite //default

01. atomic
-Atomic means only one thread access the variable(static type).
-Atomic is thread safe.
-but it is slow in performance
-atomic is default behavior
-Atomic accessors in a non garbage collected environment (i.e. when using retain/release/autorelease) will use a lock to ensure that another thread doesn't interfere with the correct setting/getting of the value.
-it is not actually a keyword.
Example :

```
@property (retain) NSString *name;

@synthesize name;
```

02. nonatomic
-Nonatomic means multiple thread access the variable(dynamic type).
-Nonatomic is thread unsafe.
-but it is fast in performance
-Nonatomic is NOT default behavior,we need to add nonatomic keyword in property attribute.
-it may result in unexpected behavior, when two different process (threads) access the same variable at the same time.

Example:

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

@synthesize name;
```

Explain:

Suppose there is an atomic string property called "name", and if you call [self setName:@"A"] from thread A, call [self setName:@"B"] from thread B, and call [self name] from thread C, then all operation on different thread will be performed serially which means if one thread is executing setter or getter, then other threads will wait. This makes property "name" read/write safe but if another thread D calls [name release] simultaneously then this operation might produce a crash because there is no setter/getter call involved here. Which means an object is read/write safe (ATOMIC) but not thread safe

as another threads can simultaneously send any type of messages to the object. Developer should ensure thread safety for such objects.

If the property "name" was nonatomic, then all threads in above example - A,B, C and D will execute simultaneously producing any unpredictable result. In case of atomic, Either one of A, B or C will execute first but D can still execute in parallel.

03. strong (iOS4 = retain)
-it says "keep this in the heap until I don't point to it anymore"
-in other words " I'am the owner, you cannot dealloc this before aim fine with that same as retain"
-You use strong only if you need to retain the object.
-By default all instance variables and local variables are strong pointers.
-We generally use strong for UIViewControllers (UI item's parents)
-strong is used with ARC and it basically helps you , by not having to worry about the retain count of an object. ARC automatically releases it for you when you are done with it.Using the keyword strong means that you own the object.

Example:

```
@property (strong, nonatomic) ViewController *viewController;

@synthesize viewController;
```

04. weak (iOS4 = unsafe_unretained)
-it says "keep this as long as someone else points to it strongly"
-the same thing as assign, no retain or release
-A "weak" reference is a reference that you do not retain.
-We generally use weak for IBOutlets (UIViewController's Childs).This works because the child object only needs to exist as long as the parent object does.
-a weak reference is a reference that does not protect the referenced object from collection by a garbage collector.
-Weak is essentially assign, a unretained property. Except the when the object is deallocated the weak pointer is automatically set to nil

Example :

```
@property (weak, nonatomic) IBOutlet UIButton *myButton;

@synthesize myButton;
```

Explain:

Imagine our object is a dog, and that the dog wants to run away (be deallocated).
Strong pointers are like a leash on the dog. As long as you have the leash attached to the dog, the dog will not run away. If five people attach their leash to one dog, (five strong pointers to one object), then the dog will not run away until all five leashes are detached.
Weak pointers, on the other hand, are like little kids pointing at the dog and saying "Look! A dog!" As long as the dog is still on the leash, the little kids can still see the dog, and they'll still point to it. As soon as all the leashes are detached, though, the dog runs away no matter how many little kids are pointing to it.
As soon as the last strong pointer (leash) no longer points to an object, the object will be deallocated, and all weak pointers will be zeroed out.
When we use weak?
The only time you would want to use weak, is if you wanted to avoid retain cycles
(e.g. the parent retains the child and the child retains the parent so neither is ever released).

05. retain = strong
-it is retained, old value is released and it is assigned
-retain specifies the new value should be sent -retain on assignment and the old value sent -release
-retain is the same as strong.
-apple says if you write retain it will auto converted/work like strong only.
-methods like "alloc" include an implicit "retain"

Example:

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

@synthesize name;
```

06. assign
-assign is the default and simply performs a variable assignment
-assign is a property attribute that tells the compiler how to synthesize the property's setter implementation
-I would use assign for C primitive properties and weak for weak references to Objective-C objects.

Example:

```
@property (nonatomic, assign) NSString *address;

@synthesize address;
```

07. unsafe_unretained

-unsafe_unretained is an ownership qualifier that tells ARC how to insert retain/release calls
-unsafe_unretained is the ARC version of assign.

Example:

```
@property (nonatomic, unsafe_unretained) NSString *nickName;

@synthesize nickName;
```

08. copy
-copy is required when the object is mutable.
-copy specifies the new value should be sent -copy on assignment and the old value sent -release.
-copy is like retain returns an object which you must explicitly release (e.g., in dealloc) in non-garbage collected environments.
-if you use copy then you still need to release that in dealloc.
-Use this if you need the value of the object as it is at this moment, and you don't want that value to reflect any changes made by other
owners of the object. You will need to release the object when you are finished with it because you are retaining the copy.

Example:

```
@property (nonatomic, copy) NSArray *myArray;

@synthesize myArray;
```

09. readonly
-declaring your property as readonly you tell compiler to not generate setter method automatically.
-Indicates that the property is read-only.
-If you specify readonly, only a getter method is required in the @implementation block. If you use the @synthesize directive in
the @implementation block, only the getter method is synthesized. Moreover, if you attempt to assign a value using the dot syntax,
you get a compiler error.

Example:

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

```
@synthesize name;
```

10. readwrite
-setter and getter generated.
-Indicates that the property should be treated as read/write.
-This attribute is the default.
-Both a getter and setter method are required in the @implementation block. If you use the @synthesize directive in the implementation
block, the getter and setter methods are synthesized.

Example:

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


@synthesize name;
```

Variable property attributes or Modifiers in iOS

Variable property attributes or Modifiers

Property Attributes Indicate Data Accessibility and Storage Considerations

Use Accessor Methods to Get or Set Property Values

01. atomic //default
02. nonatomic
03. strong=retain //default
04. weak= unsafe_unretained
05. retain
06. assign //default
07. unsafe_unretained
08. copy
09. readonly
10. readwrite //default

01. atomic

-Atomic means only one thread access the variable(static type).

-Atomic is thread safe.

-but it is slow in performance

-atomic is default behavior

-Atomic accessors in a non garbage collected environment (i.e. when using retain/release/autorelease) will use a lock to

ensure that another thread doesn't interfere with the correct setting/getting of the value.

-it is not actually a keyword.

Example :

```
@property (retain) NSString *name;
```

```
@synthesize name;
```

02. nonatomic

-Nonatomic means multiple thread access the variable(dynamic type).

-Nonatomic is thread unsafe.

-but it is fast in performance

-Nonatomic is NOT default behavior,we need to add nonatomic keyword in property attribute.

-it may result in unexpected behavior, when two different process (threads) access the same variable at the same time.

Example:

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

@synthesize name;

Explain:

Suppose there is an atomic string property called "name", and if you call [self setName:@"A"] from thread A,

call [self setName:@"B"] from thread B, and call [self name] from thread C, then all operation on different thread will be performed serially which means if one thread is executing setter or getter, then other threads will wait. This makes property "name" read/write safe but if another thread D calls [name release] simultaneously then this operation might produce a crash because there is no setter/getter call involved here. Which means an object is read/write safe (ATOMIC) but not thread safe as another threads can simultaneously send any type of messages to the object. Developer should ensure thread safety for such objects.

If the property "name" was nonatomic, then all threads in above example - A,B, C and D will execute simultaneously producing any unpredictable result. In case of atomic, Either one of A, B or C will execute first but D can still execute in parallel.

03. strong (iOS4 = retain)

-it says "keep this in the heap until I don't point to it anymore"

-in other words " I'am the owner, you cannot dealloc this before aim fine with that same as retain"

-You use strong only if you need to retain the object.

-By default all instance variables and local variables are strong pointers.

-We generally use strong for UIViewControllers (UI item's parents)

-strong is used with ARC and it basically helps you , by not having to worry about the retain count of an object. ARC automatically releases it for you when you are done with it.Using the keyword strong means that you own the object.

Example:

```
@property (strong, nonatomic) ViewController *viewController;

@synthesize viewController;
```

04. weak (iOS4 = unsafe_unretained)

-it says "keep this as long as someone else points to it strongly"

-the same thing as assign, no retain or release

-A "weak" reference is a reference that you do not retain.

-We generally use weak for IBOutlets (UIViewController's Childs).This works because the child object only

needs to exist as long as the parent object does.

-a weak reference is a reference that does not protect the referenced object from collection by a garbage collector.

-Weak is essentially assign, a unretained property. Except the when the object is deallocated the weak pointer is automatically set to nil

Example :

```
@property (weak, nonatomic) IBOutlet UIButton *myButton;

@synthesize myButton;
```

Explain:

Imagine our object is a dog, and that the dog wants to run away (be deallocated).

Strong pointers are like a leash on the dog. As long as you have the leash attached to the dog, the dog will not run away. If five people attach their leash to one dog, (five strong pointers to one object), then the dog will not run away until all five leashes are detached.

Weak pointers, on the other hand, are like little kids pointing at the dog and saying "Look! A dog!" As long as the dog is still on the leash, the little kids can still see the dog, and they'll still point to it. As soon as all the leashes are detached, though, the dog runs away no matter how many little kids are pointing to it.

As soon as the last strong pointer (leash) no longer points to an object, the object will be deallocated, and all weak pointers will be zeroed out.

When we use weak?

The only time you would want to use weak, is if you wanted to avoid retain cycles

(e.g. the parent retains the child and the child retains the parent so neither is ever released).

05. retain = strong

-it is retained, old value is released and it is assigned

-retain specifies the new value should be sent -retain on assignment and the old value sent -release

-retain is the same as strong.

-apple says if you write retain it will auto converted/work like strong only.

-methods like "alloc" include an implicit "retain"

Example:

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

```
@synthesize name;
```

06. assign

-assign is the default and simply performs a variable assignment

-assign is a property attribute that tells the compiler how to synthesize the property's setter implementation

-I would use assign for C primitive properties and weak for weak references to Objective-C objects.

Example:

```
@property (nonatomic, assign) NSString *address;

@synthesize address;
```

07. unsafe_unretained

-unsafe_unretained is an ownership qualifier that tells ARC how to insert retain/release calls

-unsafe_unretained is the ARC version of assign.

Example:

```
@property (nonatomic, unsafe_unretained) NSString *nickName;

@synthesize nickName;
```

08. copy

-copy is required when the object is mutable.

-copy specifies the new value should be sent -copy on assignment and the old value sent -release.

-copy is like retain returns an object which you must explicitly release (e.g., in dealloc) in non-garbage collected environments.

-if you use copy then you still need to release that in dealloc.

-Use this if you need the value of the object as it is at this moment, and you don't want that value to reflect any changes made by other

owners of the object. You will need to release the object when you are finished with it because you are retaining the copy.

Example:

```
@property (nonatomic, copy) NSArray *myArray;
```

```
@synthesize myArray;
```

09. readonly

-declaring your property as readonly you tell compiler to not generate setter method automatically.

-Indicates that the property is read-only.

-If you specify readonly, only a getter method is required in the @implementation block. If you use the @synthesize directive in

the @implementation block, only the getter method is synthesized. Moreover, if you attempt to assign a value using the dot syntax,

you get a compiler error.

Example:

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

@synthesize name;
```

10. readwrite

-setter and getter generated.

-Indicates that the property should be treated as read/write.

-This attribute is the default.

-Both a getter and setter method are required in the @implementation block. If you use the @synthesize directive in the implementation

block, the getter and setter methods are synthesized.

Example:

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

@synthesize name;
```

iPhone Interview Questions And Answers For Freshers

What is iPhone?

IPhone is a combination of internet and multimedia enabled smart phone developed by Apple Inc.IPhone functions as a camera phone, including text messaging, and visual voice mail. IPhone is a portable media player that resembles a video iPod It has user interface that is built around the multi-touch screen including virtual keyboard.

What is an iPhone app?

An iPhone app is a program that runs on our iPhone/iPod Touch. It enables us to ccomplish a certain task. They could be utility apps, games, enterprise apps, entertainment apps, apps to access our bank account etc.

Introduction to Iphone application Development?

In 2007, Apple entered the cellular phone business with the introduction of the iPhone, a multi-touch display cell phone, which also includes the features of iPod.

Multitasking support is available from which version?

iOS 4.0

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

256bytes.

What are the features of iPhone 3gs?

- Video: Videos can be edited, shared. High quality VGA video can be shot in portrait or landscape.
- 3 Mega pixel Camera: Still photos with greater quality can be taken
- Voice control: It recognizes the names in contacts and recognizes the music on iPod.
- Compass: iPhone 3GS has built-in digital compass, used to point the way.
- Internet Tethering: Internet surfing can be done from anywhere. A 3G connection can be shared on Iphon3 with Mac notebook or laptop.

Why iPhone apps are popular?

Give our business a whole new way of transacting business for millions of users.

IPhones are the market leaders in the smart phone segment. The iPhone has become a great device to surf the internet, play games, interact with social networks and transact business.

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.

Does iOS support multitasking?

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.

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.

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.

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.

What are the requirements for developing iPhone Apps?

Mac OS 10.5/10.6 iPhone SDK (Software Development Kit 3.0/4.0).

IPhone SDK consists of:

- IDE to develop iPhone Apps is XCode(This tool is inbuilt in iPhone SDK)
- **Interface Builder** This is used to design GUI of Apps(Inbuilt feature of iPhone SDK)
- **Instruments** This is used to check any memory leaks in our apps (Inbuilt in SDK)
- **Simulator** This is used to test our apps before deploying into real device.

What are the popular apps of iPhone?

- Face book-Social networking
- Doodle Buddy-drawing
- Pandora Radio-radio on our iPhone
- Yelp-restaurant reviews

What is iPhone reference library?

iPhone reference library is a set of reference documents for iPhone OS. It can be downloaded by subscribing to the iPhone OS Library doc set. Select Help>Documentation from X code, and click the subscribe button next to the iPhone OS Library doc set, which appears in the left column.

What is iPhone sdk?

iPhone SDK is available with tools and interfaces needed for developing, installing and running custom native applications. Native applications are built using the iPhone OS's system frameworks and Objective-C language and run directly on iPhone OS. Native applications are installed physically on a device and can run in presence or absence of network connection.

What is iPhone architecture?

It is similar to Mac OS 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.

What are the location services?

Applications such as Maps, camera and compass are allowed to use the information from cellular, Wi-Fi and Global Positioning System networks for determining the approximate locations.

The location is displayed on the screen, using a blue marker.

Describe the functionality of accelerometer of an iPhone ?

iPhone responds to motion using a built-in accelerometer.

The accelerometer detects the movement and changes the display accordingly, at the time of rotating iPhone from portrait to landscape.

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.

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.

Explain about the applications that can be used with iPhone ?

Technology, Entertainment and Design (TED): Allows to watch and listen to world's most fascinating people have to say, all on the iPhone.

What are the tools required to develop iOS applications?

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

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

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.

What is iPhone reference library?

iPhone reference library is a set of reference documents for iPhone OS.

It can be downloaded by subscribing to the iPhone OS Library doc set.

Select Help>Documentation from X code, and click the subscribe button next to the iPhone OS Library doc set, which appears in the left column.

What are sensors in iPhone?

The proximity sensor immediately turns off the display when the iPhone is lifted to ear. With this sensor the power is saved and accidental dialing is prevented.

The display is automatically brightens the iPhone by the ambient light sensor when the sunlight or bright rooms and dims in darker places.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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

Background state.

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.

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.

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.

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

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

What happens if IApplication object does not handle an event?

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

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.

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.

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.

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.

You wish to define your custom view. Which class will be subclassed?

Custom views can be defined by subclassing UIView.

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.

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.

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.

What is the purpose of UIWindow object?

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

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.

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.

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.

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.To achieve this, Person Object must register as an observer of the BankAccount’s accountBalance property by sending an

addObserver:forKeyPath:options:context: message.

What is difference between NSNotification and delegate?

Delegate is passing message from one object to other object. It is like one to one communication while nsnotification is like passing message to multiple objects at the same time. All other objects that have subscribed to that notification or acting observers to that notification can or can’t respond to that event. Notifications are easier but you can get into trouble by using those like bad architecture. Delegates are more frequently used and are used with help of protocols.

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).

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.

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.

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.)

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.

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).

What is iPhone?

IPhone is a combination of internet and multimedia enabled smart phone developed by Apple Inc.....

What are the features of iphone 3gs?

Video: Videos can be edited, shared. High quality VGA video can be shot in portrait or landscape.....

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.....

What is iphone sdk?

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

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.....

What is iphone reference library?

iPhone reference library is a set of reference documents for iPhone OS. It can be downloaded by subscribing to the iPhone OS Library doc set.....

What are sensors in iphone?

The proximity sensor immediately turns off the display when the iPhone is lifted to ear. With this sensor the power is saved and accidental dialing is prevented.....

What are the location services?

Applications such as Maps, camera and compass are allowed to use the information from cellular, Wi-Fi and Global Positioning System.....

Describe the functionality of accelerometer of an iphone

iPhone responds to motion using a built-in accelerometer.....

Explain about the applications that can be used with iphone.

Technology, Entertainment and Design(TED) : Allows to watch and listen to world's most fascinating people have to say, all on the iPhone.....

Important Questions And Answers

1.What is latest iOS version?

IOS - 6.1.3 (updated on 5/15/13 3:15 AM Pacific Daylight Time)

2.What is latest Xcode version?

Xcode- 4.6.2 (updated on 5/15/13 3:15 AM

Pacific Daylight Time)

3.What is latest mac os version?

Mac- Mountain Lion (updated on 5/15/13 3:15 AM

Pacific Daylight Time)

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.

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) );
```

```
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.

1. .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. 2. Minimizes 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 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

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 coredata & sqlite?

There is a huge difference between these two. SQLLite 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 dont 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 coredata?

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

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:toProposedIndexPath:

Copying and Pasting Row Content

– tableView:shouldShowMenuForRowAtIndexPath:

– tableView:canPerformAction:forRowAtIndexPath:withSender:

– tableView:performAction:forRowAtIndexPath:withSender:

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 -

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:expectedTotalBytes:

– 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:notationName:

– 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