# Lập trình trên iOS

Fujinet seminar, May 2012

# Outline

- Giới thiệu chung về iOS
- Khái quát về ngôn ngữ lập trình Objective-C
- Lập trình trên iOS
- Công cụ lập trình & case study

# Giới thiệu iOS

- Sơ lược về iOS
- Lập trình trên iOS

# Sơ lược iOS

May, 2012

- 250,000,000 iOS devices sold
- 645,669 Apps
- 898 Apps / Day
- 146,230 Publisher
- 25,000,000,000 downloads



Source : http://148apps.biz/app-store-metrics/,

# Kiếm tiền trên iOS: Paid vs. Free

Ex: Game Doodle Jump (công ty 2 người Lima Sky) : 2.08 triệu $ / year


quảng cáo (iAds, Google Adsense, Admob, ...)


Ex: Free Angry Bird : over 1 triệu $ / month from ads each platform (android, iOS)

# Lập trình trên iOS

- Quy trình phát triển ứng dụng trên iOS
- The iOS developer tools
- The iOS platform
- The Objective-C language

# Quy trình lập trình trên iOS

## iOS

**iOS Developer Program**
Individual
$99 / Year

For an individual developer who will be creating free and commercial iOS apps for distribution on the App Store.

**iOS Developer Program**
Company
$99 / Year

For a company with a development team who will be creating free and commercial iOS apps for distribution on the App Store.

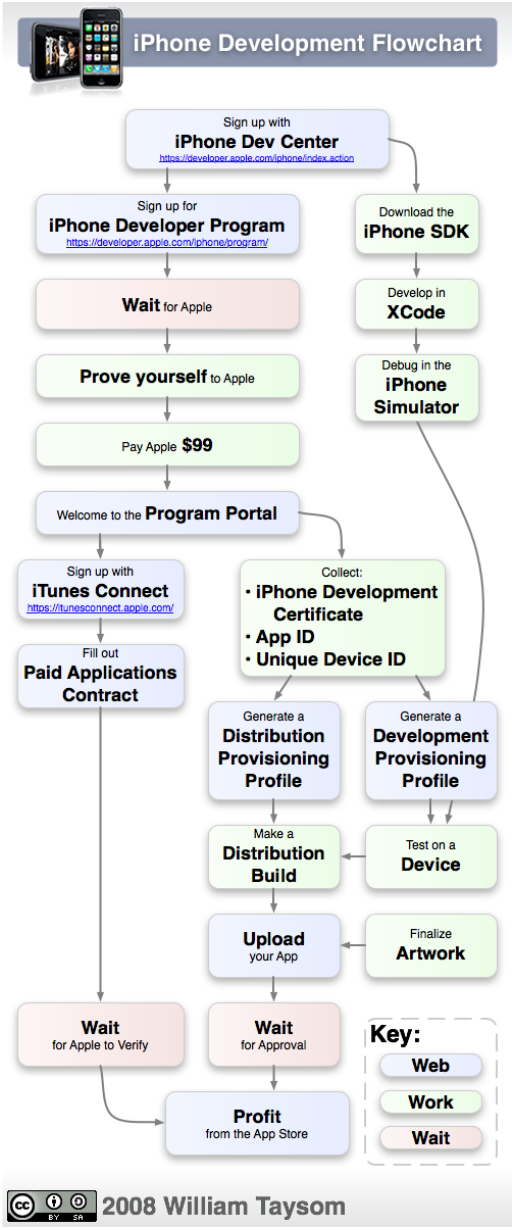
**iOS Developer Enterprise Program**
$299 / Year

For a company who will be creating proprietary, in-house iOS apps.

**Note**: A Dun & Bradstreet Number is required.

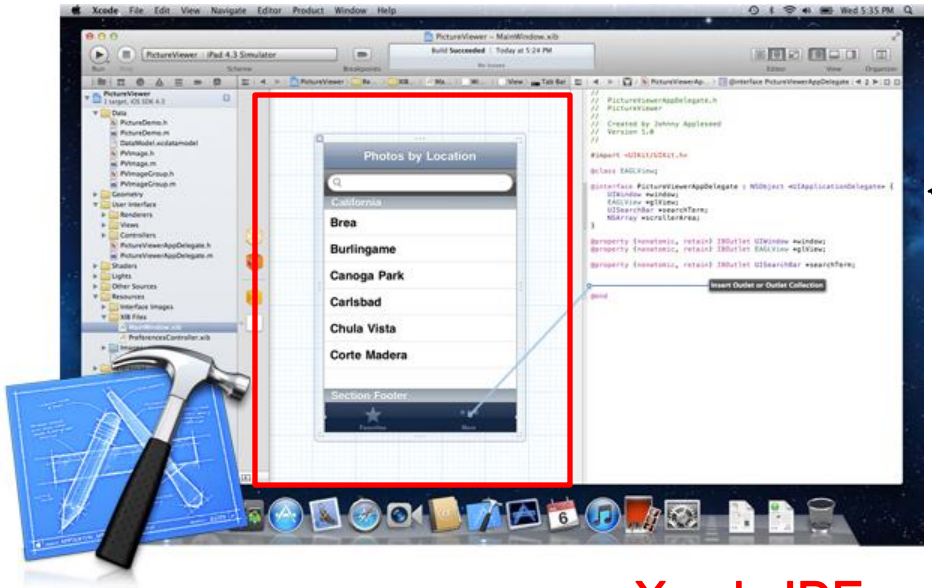
**iOS Developer University Program**
Free

For higher education institutions looking to introduce iOS development into their curriculum.




Source: http://www.oziphonepro.com/2009/04/iphone-app-development-flowchart.html

# The iOS developer tools




Interface Builder

Xcode IDE

Performance

Instruments

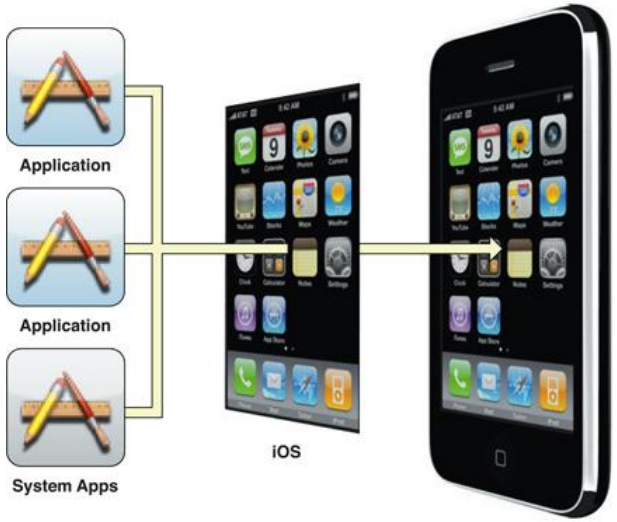
Run, Debug

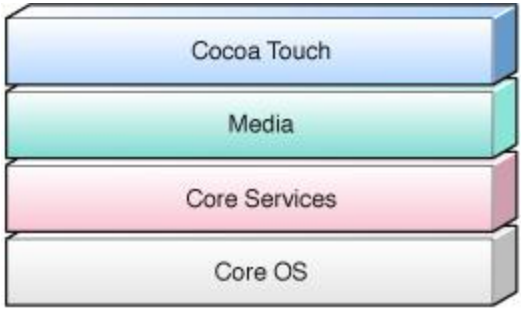
Simulator

More reading:
http://developer.apple.com/library/ios/#referencelibrary/GettingStarted/URL_Tools_for_iPhone_OS_Development/_index.html

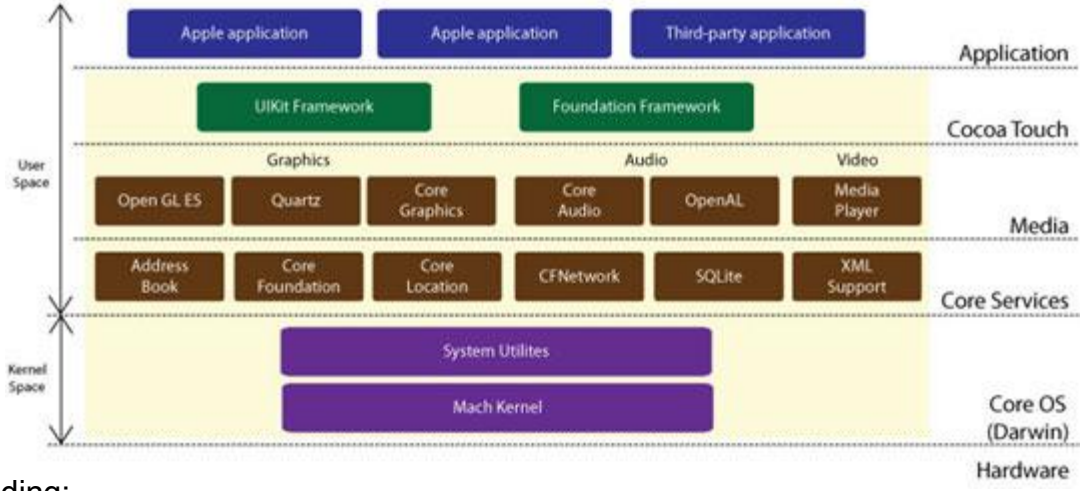
# The iOS platform

Applications layered on top of iOS




Layers of iOS





More reading:
http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html

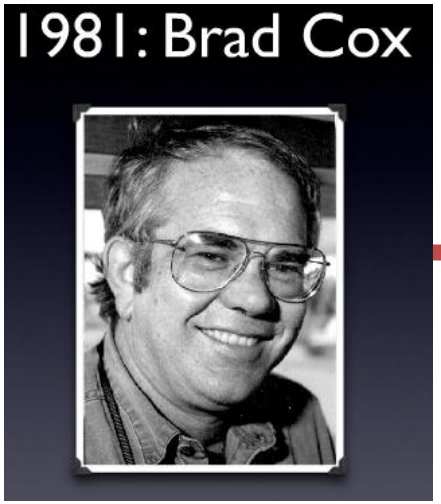
# Objective-C

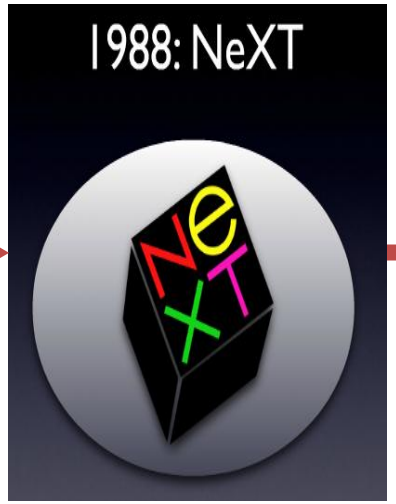
- Ngôn ngữ chính để phát triển trên iOS
- Ngôn ngữ hướng đối tượng

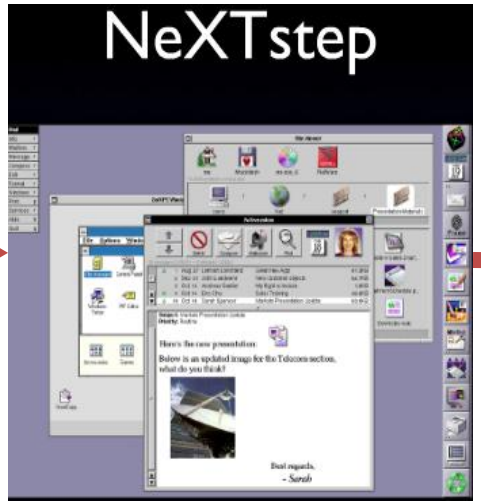
  (OOP — Object-oriented programming)

  - **Class:** A class defines the grouping of data and code (or action). It is the "type" of an object.
  - **Instance:** a specific allocation of a class in memory. Instance variables are typically hidden, and they are accessed by getter/setter method.
  - **Method:** behavior, operations (coding as subroutine, function) applied to instances.
  - **Instance Variable:** attributes, data belonging to an instance.

# Lịch sử phát triển Objective-C






# Một số điểm trong Objective-C

- Class, instance, property
- Syntax
- Inheritance (kế thừa)
- Data type & Operators
- Selectors and Targets
- Structures & Flow Control Statements
- Memory Management
- The Model-View-Controller Design Pattern

# Class, instance & property

```objc
//  Car.h
#import <Foundation/Foundation.h>

@interface Car : NSObject {
    NSString *color;
    float displacement;
    float speed;
    int direction;
}
@property (nonatomic, retain) NSString *color;
@property (nonatomic, assign) float speed;

- (void) drive:(float)drive_speed withDir: (int) dir;
- (void) changeDirection: (int) newDirection;
@end
```

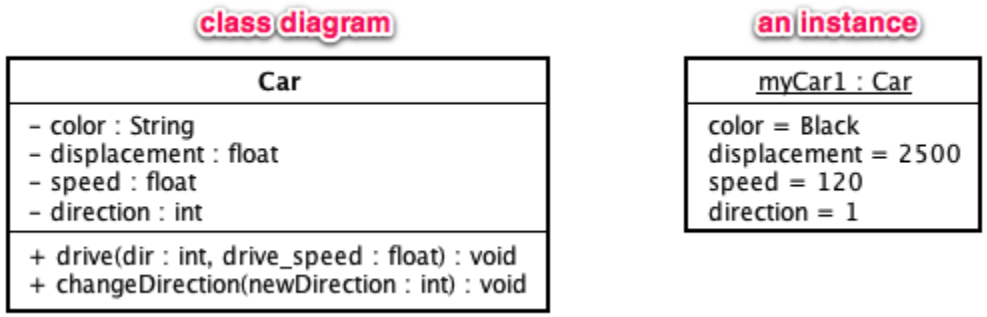
```objc
//  Car.m
#import "Car.h"

@implementation Car
@synthesize color, speed;

- (id)init{
    speed = 0;
    direction = 1;
    return self;
}
```

```objc
//example.m
Car *aCar;
aCar = [[Car alloc] init];
aCar.speed = 90 ;
```

# Class, instance & property



# Property Attributes

| Attribute | Description |
|---|---|
| Read/Write attributes. The default setting for a property is readwrite. | |
| readwrite | Tells the compiler the property may be read and written to. |
| readonly | Tells the compiler that the property may only be read. |
| Setter attributes. The default setting for a property is assign. | |
| assign | Tells the compiler that the property uses assignment in the property's setter. |
| retain | Tells the compiler that the property's setter should call `retain` on the property when setting it. |
| copy | Tells the compiler that the property's setter should create a new copy of the object. |
| Other property attributes. | |
| nonatomic | Tells the compiler that the property's accessor methods are not thread safe. |
| setter= | Tells the compiler that you wish specifying a setter of your own name. |
| getter= | Tells the compiler that you wish specifying a getter of your own name. |

# Syntax

- **Message syntax**

  Định dạng của message syntax là [receiver message].

  Ví dụ: [aCar color];  // Gọi method “getColor” của instance “aCar”

  Method “getColor” tự sinh khi định nghĩa property

  Method này còn gọi là message

  Message syntax có thể chứa tham số (arguments).

  [receiver message: argument] hoặc  [receiver message: arg1 andArg: arg2];

  Ví dụ:  [aCar drive:90 withDir:2];

- **Dot syntax.**

  Ví dụ: aCar.color = @”red”; // Gọi method “setColor” của instance “aCar”

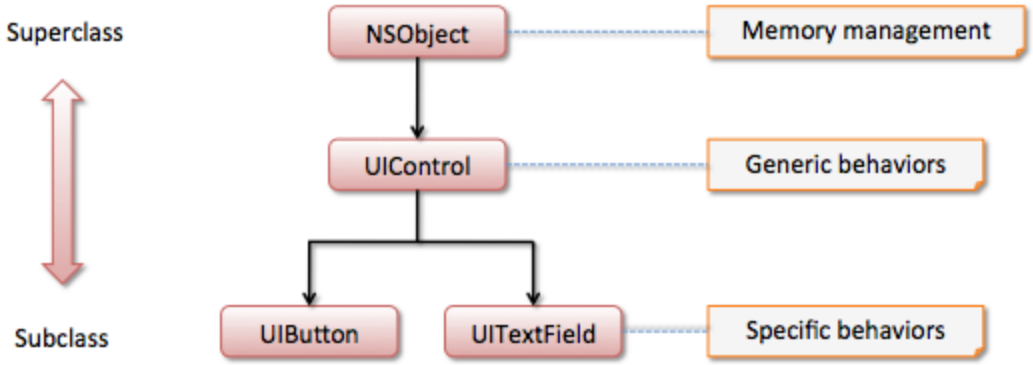
Ví dụ: NSLog(@”%@, %d”, [aCar color], aCar.speed“);

// Gọi method “getColor” và “getSpeed” của instance “aCar”

# Inheritance (kế thừa)

- Subclass kế thừa data và method từ superclass

- Các đặc điểm khác như **NO** overload, overriding cũng giống các ngôn ngữ hướng đối tượng khác (như Java)



# Data types & Operators

- Giống ngôn ngữ lập trình C như int, float, char, +, -, x, /
- Các kiểu dữ liệu tiêu biểu khác
  - Dynamic and static object:
    id anInstance; // dynamically-typed object
    Car *aCar; // statically-typed object
  - NULL object: **nil**.
    myCar = nil;
    if (myCar == nil) return;
  - **BOOL**.
    BOOL isRunning = YES;
    if (!isRunning) return;
  - **NSString**
    NSString *aStr = @“Hello World!”;
    NSString *aStr1 = [NSString stringWithFormat:@"Object: %@, speed=%d", aCar.color, aCar.speed];

# Data types & Operators

- Các kiểu dữ liệu tiêu biểu khác (cont)
  - **NSMutableString**.
    NSMutableString *newStr = [NSMutableString string];
    [newStr appendString:@"-- tail"];
  - **NSArray**.
    NSArray *ia = [NSArray arrayWithObjects:aCar, nil];
    Car *tCar = [ia objectAtIndex:0];
    float s = tCar.speed;
  - **NSMutableArray**.
    NSMutableArray *ia = [NSMutableArray arrayWithObjects:aCar, nil];
    [ia addObject:bCar];
    [ia removeObjectAtIndex:1];
  - **NSDictionary**, **NSMutableDictionary**.
    NSMutableDictionary *dic = [NSMutableDictionary dictionary];
    [dic setObject:aCar forKey:@"red-Car"];
    Car *tCar = [dic valueForKey:@"red-Car"];
  - **NSNumber**.
    // num is a pointer
    NSNumber *num = [NSNumber numberWithInt:2];

# Selectors and Targets

- Dynamically call at runtime

- Selector: SEL.

```
// selector: the method name (function pointer)
SEL sel = @selector(disable_buttonB:);
[buttonA performSelctor:sel withObject:self];
```

- Target-Action pattern

```
- (void)viewDidLoad {
    button = [[UIBarButtonItem alloc]
                initWithTitle:@"Done"
                        style: UIBarButtonItemStyleBordered
                       target:self
                       action:@selector(doneButtonHit:)];
}
```

# Structures & Flow Control Statements

- Giống ngôn ngữ lập trình C
  - Structures
    - **CGPoint**

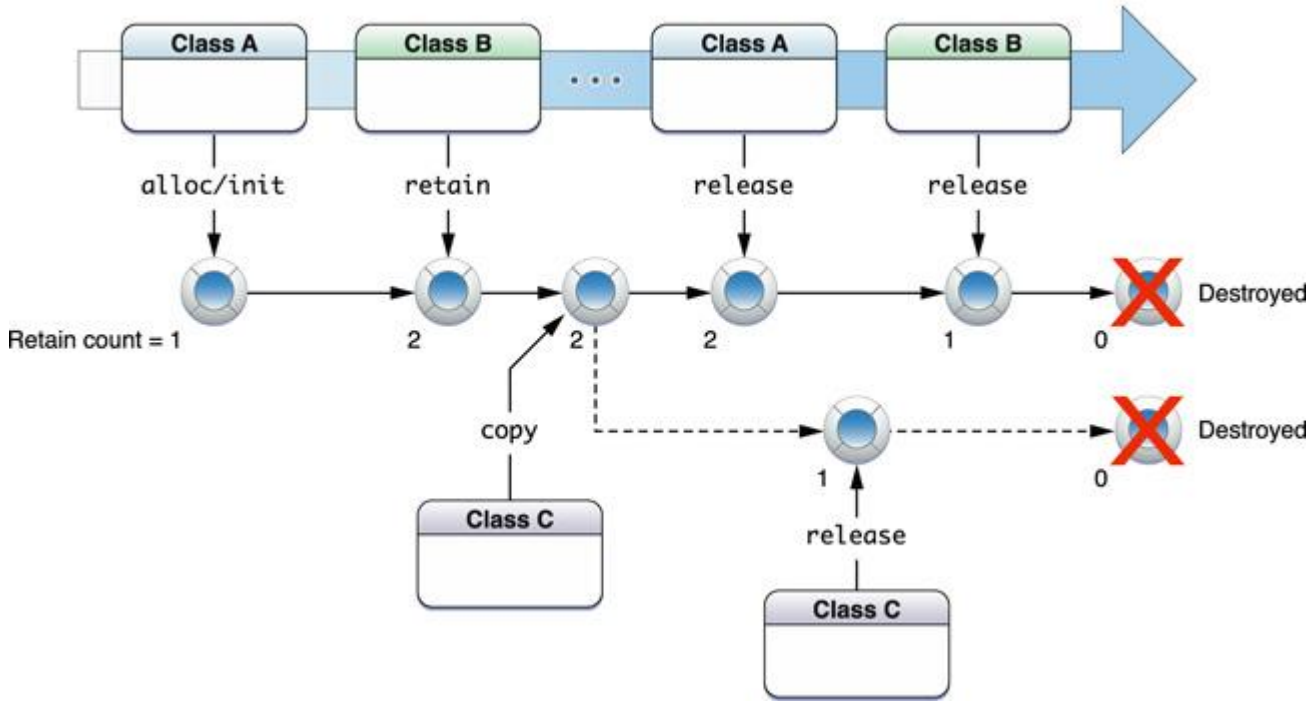
```
struct CGPoint {
    CGFloat x;
    CGFloat y;
};
typedef struct CGPoint CGPoint;
```

  - Flow Control Statements

```
int x = 10;
if ( x > 9 ) {
    NSLog (@"x is greater than 9!");
} else {
    NSLog (@"x is less than 9!");
}
```

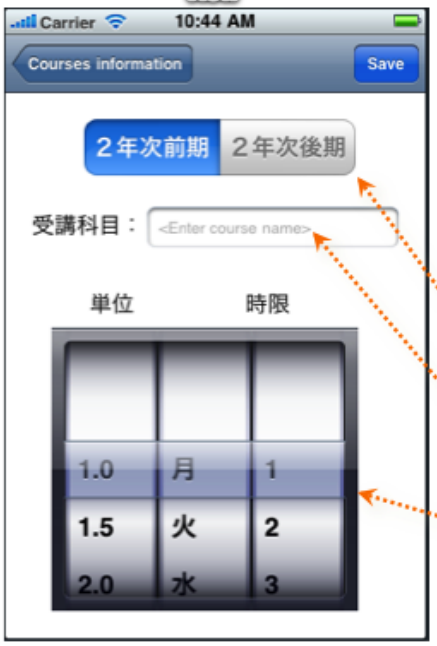
# Memory Management

- No garbage collection
- ARC
- Autorelease and pool



# The Model-View-Controller

View




Controller

**NewCourseViewController**

```
IBOutlet UISegmentedControl *segmentedControl;
IBOutlet UITextField *textField;
IBOutlet UIPickerView *pickerView;
NSArray *credits;
NSArray *week;
NSArray *time;
```

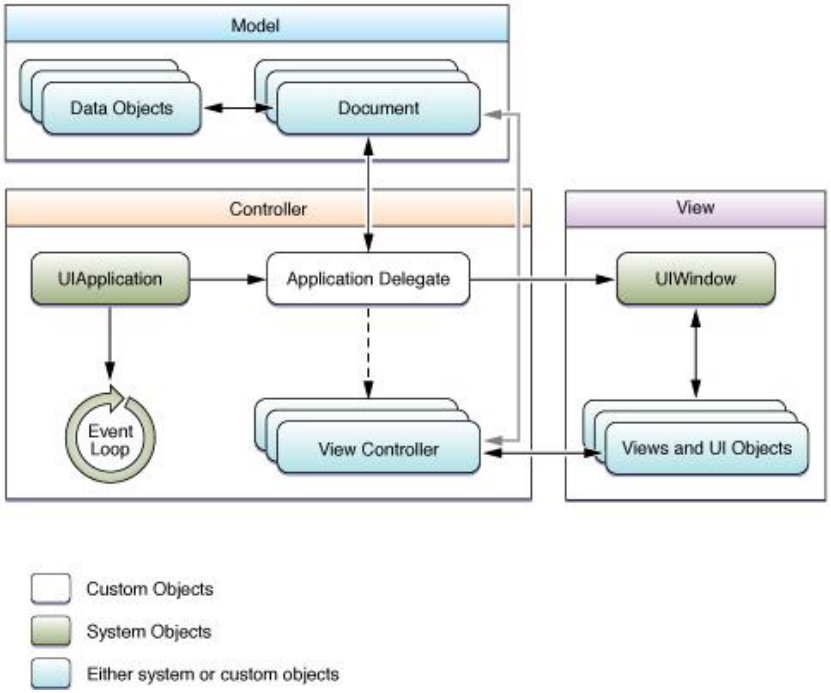
```
-(void) saveItem:(id)sender;
```

Model

**CoursesData**

```
NSMutableArray *groupNames;
NSMutableArray *courseNames;
NSMutableArray *credits;
NSMutableArray *time;
NSMutableArray *dataArray;
```

```
-(id) initWithData;
-(void) parseData:(NSArray *)array;
- (int)countOfGroups;
- (int)countOfCoursesInGroup:(int)group;
- (void) addNewCourse: (NSMutableArray *)cdataArray;
- (void) deleteCourse: (NSIndexPath *)indexPath;
```



# Lập trình trên iOS

- UIControl → UIView → UITableView
  → UIButton
  → UIWebView (webkit, javasript)
- Touches
- Animation
- Webkit, javascript
- iOS SDK library
- Well-known open source library
  - Three20 (Facebook)
  - Cocos2d

More reading:
https://developer.apple.com/library/ios/#samplecode/UICatalog/Introduction/Intro.html#//apple_ref/doc/uid/DTS40007710

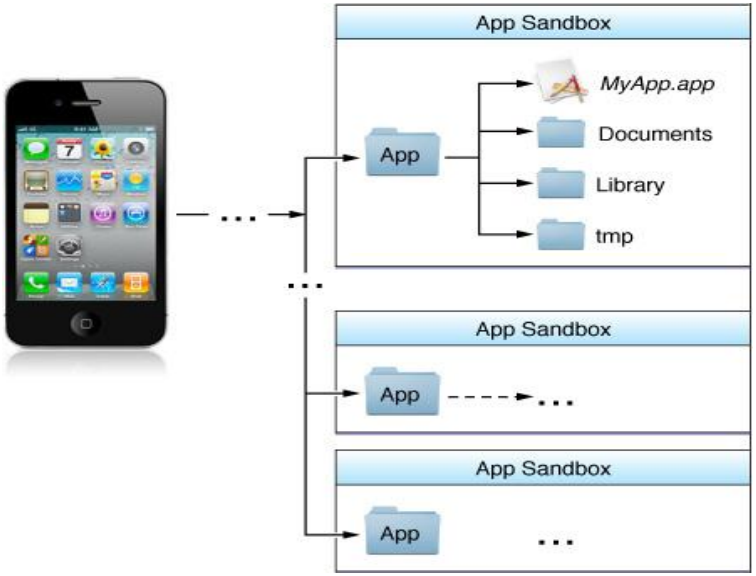
# Touches

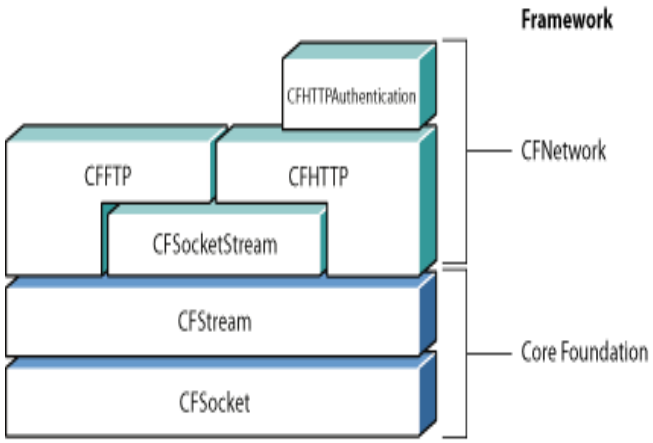
- Touch event handle
  - (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event;
  - (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event;
  - (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event;
  - (void)touchesCancelled:(NSSet *)touches withEvent:(UIEvent *)event
- Gesture Recognizers

| **Tapping (any number of taps)** | **UITapGestureRecognizer** |
|---|---|
| Pinching in and out (for zooming a view) | UIPinchGestureRecognizer |
| Panning or dragging | UIPanGestureRecognizer |
| Swiping (in any direction) | UISwipeGestureRecognizer |
| Rotating (fingers moving in opposite directions) | UIRotationGestureRecognizer |
| Long press (also known as “touch and hold”) | UILongPressGestureRecognizer |

# Các vấn đề khác

- Persistent Storage
  - XML
    - NSXML
    - libxml
  - SQLite
    - Core Data
  - User Defaults
    - User's prefrences
    - Setting bundle
- Basic Networking
- Threads
- Document Types

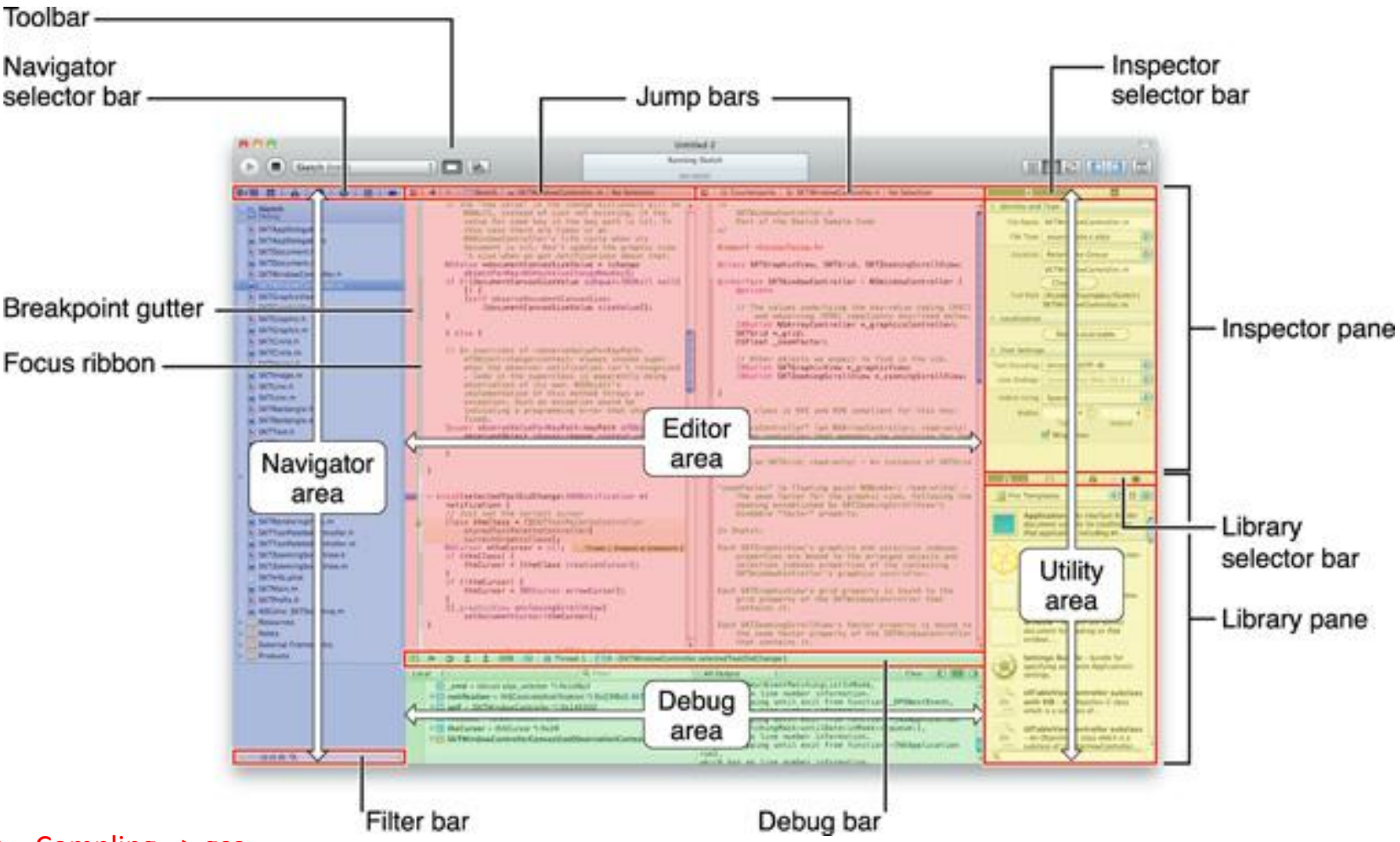


# System applications trên iOS SDK

- Photo
- Address book
- Calendar
- Mails
- Audio
- Movie

# Công cụ lập trình Xcode




- Compling -> gcc
- Linking -> ld
- Debugging -> gdb
- SCM -> svn, cvs, git

# Case study

# Summary

Đã giới thiệu những kiến thức căn bản về

- iOS
- ngôn ngữ lập trình Objective-C
- lập trình trên iOS