Xcode Lab 1
CS 274 iPhone App Development
Robert Muller

Today

• Overview of Xcode and the iPhone Simulator

• An Annotated Application

Logistics

• If you are using one of the lab systems, you’ll need to store your files on the network disk rather than the local disk;

• Lab system or laptop, create a cs274 folder in a convenient place and store your project folders there.
A Tour of Xcode

Annotated Code

```c
//main.m
//ButtonFun
//Created by Robert Mueller on 1/21/10.
//Copyright Boston College 2010. All rights reserved.
#import <UIKit/UIKit.h>

int main(int argc, char *argv[]) {
    NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
    int retVal = UIApplicationMain(argc, argv, nil, nil);
    [pool release];
    return retVal;
}
```
# Implementation Declaration

```c
// main.m
// ButtonFun
// Created by Robert Muller on 1/21/10.
// Copyright Boston College 2010. All rights reserved.
//
// Import <UIKit/UIKit.h>

int main(int argc, char *argv[]) {
    NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
    int retVal = UIApplicationMain(argc, argv, nil, nil);
    [pool release];
    return retVal;
}
```

---

# Not exactly...

```c
// main.m
// ButtonFun
// Created by Robert Muller on 1/21/10.
// Copyright Boston College 2010. All rights reserved.
//
// Import <UIKit/UIKit.h>

int main(int argc, char *argv[]) {
    NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
    int retVal = UIApplicationMain(argc, argv, nil, nil);
    [pool release];
    return retVal;
}
```

---

# Retrieving the declarations in the UIKit.h header file.

In C, header files contain declarations or specifications.

```c
// main.m
// ButtonFun
// Created by Robert Muller on 1/21/10.
// Copyright Boston College 2010. All rights reserved.
//
// Import <UIKit/UIKit.h>

int main(int argc, char *argv[]) {
    NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
    int retVal = UIApplicationMain(argc, argv, nil, nil);
    [pool release];
    return retVal;
}
```
The local variable pool will contain the address of (or point to) an instance of the built-in class NSAutoreleasePool.

A message send. This sends the message alloc to the class NSAutoreleasePool. In response, the class will return an (uninitialized) object.

Another message send. This sends the message init to an object that is an instance of class NSAutoreleasePool. Alloc-init is a ubiquitous idiom in Objective-C.
Calls the function UIApplicationMain.
UIApplicationMain creates your one and only instance of the class UIApplication.
This is your app!

int main(int argc, char *argv[])
{
    NSAutoreleasePool *pool = [NSAutoreleasePool alloc] init);
    int retVal = UIApplicationMain(argc, argv, nil, nil);
    [pool release];
    return retVal;
}

All done. Tell pool to tell the run-time system that it can reclaim the space allocated to the autorelease pool. (More on this later.)

int main(int argc, char *argv[])
{
    NSAutoreleasePool *pool = [NSAutoreleasePool alloc] init);
    int retVal = UIApplicationMain(argc, argv, nil, nil);
    [pool release];
    return retVal;
}

NB: This is a function implementation (definition) and not a method implementation.
Delegation

- Objects of a given class can “delegate” tasks to objects of a so-called “delegate” class.

- UIApplication delegates much of it’s work to an application delegate.

```swift
// ButtonFunAppDelegate.h
// ButtonFun
// Created by Robert Muller on 1/21/10.
// Copyright Boston College 2010. All rights reserved.
//
// @interface ButtonFunAppDelegate : NSObject <UIApplicationDelegate>
// @property (nonatomic, retain) IBOutlet UIWindow *window;
// @property (nonatomic, retain) IBOutlet ButtonFunViewController *viewController;
@end
```

A header file. I.e., a specification of the ButtonFunAppDelegate.
Means forward, i.e., you (the compiler) will find out about ButtonFunViewController later.

Means class. Go figure.

Means subclass. In Objective-C, every class is a subclass of NSObject.
The name in the <...> identifies a protocol. This asserts that the class ButtonFunAppDelegate implements the UIApplicationDelegate protocol.

Instances of class ButtonFunAppDelegate will have (at least) these two instance variables.

window is an instance variable of type pointer to UIWindow. This means it will have the address of an instance of the UIWindow class. (This will be the one and only window for our app.)
This is a **macro**. It tells the compiler that in the implementation file it will be asked to synthesize **setter** and **getter** methods for the declared instance variable. (More on this later.)

```c
// ButtonFunAppDelegate.h
// ButtonFun
// Created by Robert Muller on 1/21/10.
// Copyright Boston College 2010. All rights reserved.
// @property (nonatomic, retain) IBOutlet UIWindow *window;
@end
```

This is a special gizmo that tells the compiler that the instance variable will be connected to a data structure made by the Interface Builder.

```c
// ButtonFunAppDelegate.h
// ButtonFun
// Created by Robert Muller on 1/21/10.
// Copyright Boston College 2010. All rights reserved.
// @property (nonatomic, retain) IBOutlet UIWindow *window;
@end
```

NB: Although this particular class doesn’t explicitly declare any methods or functions, instances of the class will have many methods.

```c
// ButtonFunAppDelegate.h
// ButtonFun
// Created by Robert Muller on 1/21/10.
// Copyright Boston College 2010. All rights reserved.
// @property (nonatomic, retain) IBOutlet UIWindow *window;
@end
```
Synthesize the setter and getter methods specified in the matching .h file.

The implementation of a method.
The "-" annotation signifies an object method rather than a class method.
// ButtonFunAppDelegate.m
// ... comments elided...
#import "ButtonFunAppDelegate.h"
#import "ButtonFunViewController.h"

@implementation ButtonFunAppDelegate

@synthesize window;
@synthesize viewController;

- (void)applicationDidFinishLaunching:(UIApplication *)application
// Override point for customization after app launch
[window addSubview:viewController.view];
[window makeKeyAndVisible];

- (void)dealloc{
    [viewController release];
    [super dealloc];
}
@end