



Computing & Information Sciences
FLORIDA INTERNATIONAL UNIVERSITY

Mobile Application Development

lecture3

Fall 2011 - COP 4655 U1

T/R 5:00 - 6:15pm – ECS 134

Steve Luis

Agenda

- Distribute equipment and provisioning account
- A look at Objective-C Classes
- Building a View based Controller
- Programming Assignment



Provisioning Process

- Check email for an Apple invitation to join FIU Team.
- Click the link and login with the newly created Apple Developer account you created with your FIU email address.
- Open Organizer Window and select device.
- Plug in the iPad. An admin may need to authenticate.
- Select Provisioning Profile on left Navigator.
- Select the “Refresh” button at the bottom.
- You should receive your Developer and Provisioning Profile
- Select Export to save a copy to your documents directory and copy to your personal Mac if you are going to do development at home.

Review: Hello World in Objective-C

```
// main.m

#import <Foundation/Foundation.h>

int main (int argc, const char * argv[])
{

    NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];

    // insert code here...
    NSLog(@"Hello, World!");

    [pool drain];
    return 0;
}
```



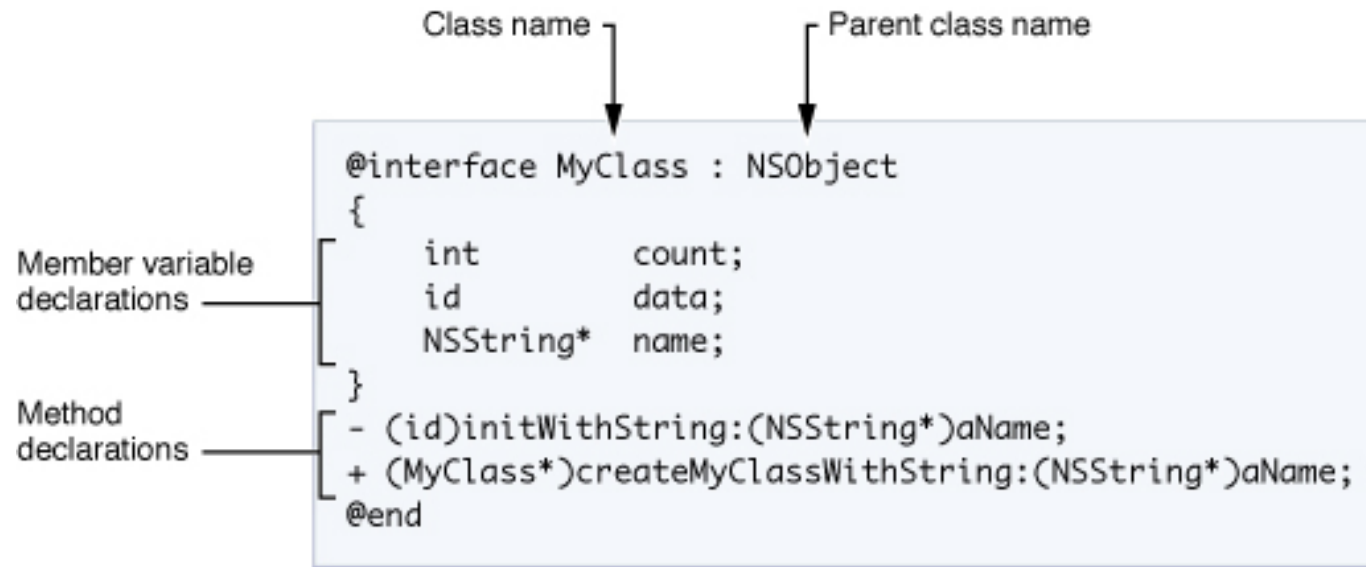
- XCODE example: Build Hello World via Project Templates/Command Line

Class Interface and Implementation

- A class interface, where you declare member variables and methods is defined in a *.h file.
 - MyFactory.h
- A class implementation, where you write the code for the methods are located in a *.m file
 - MyFactory.m
- *One class per .h/.m file.*
- To include class file use:
 - #import "Myfactory.h"
- To include class library file use:
 - #import <Foundation/Foundation.h>



Classes



http://developer.apple.com/library/mac/#referencelibrary/GettingStarted/Learning_Objective-C_A_Primer/_index.html

Class Method Definitions

```
@interface classname : superclassname {  
    // instance variables  
}  
+ classMethod1;  
  
+ (return_type)classMethod2;  
  
+ (return_type)classMethod3:(param1_type)param1_varName;  
  
- (return_type)instanceMethod1:(param1_type)param1_varName  
    :(param2_type)param2_varName;  
  
- (return_type)instanceMethod2WithParameter:(param1_type)param1_varName  
    andOtherParameter:(param2_type)param2_varName;  
  
@end
```

Note

- initWithObject:object { ... }

is the same as

- (id)initWithObject:(id)object { ... }

Implementation

```
#import "ClassName.h"
```

```
@implementation ClassName
```

```
method definitions
```

```
@end
```

Implementing Methods

```
+ (id)alloc {
```

```
  ...
```

```
}
```

```
- (BOOL)isFilled {
```

```
  ...
```

```
}
```

Object Messaging

[receiver message];

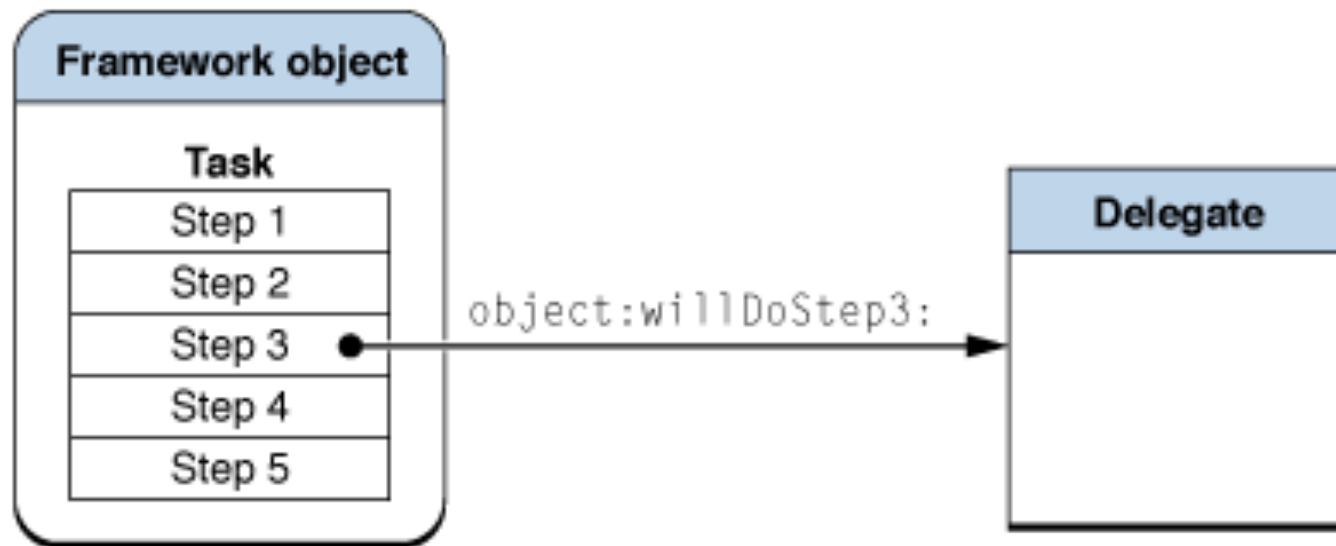
[myRectangle draw];

[myRectangle size: 5.0];

[myRectangle setOriginX: 3.0 y: 2.0];

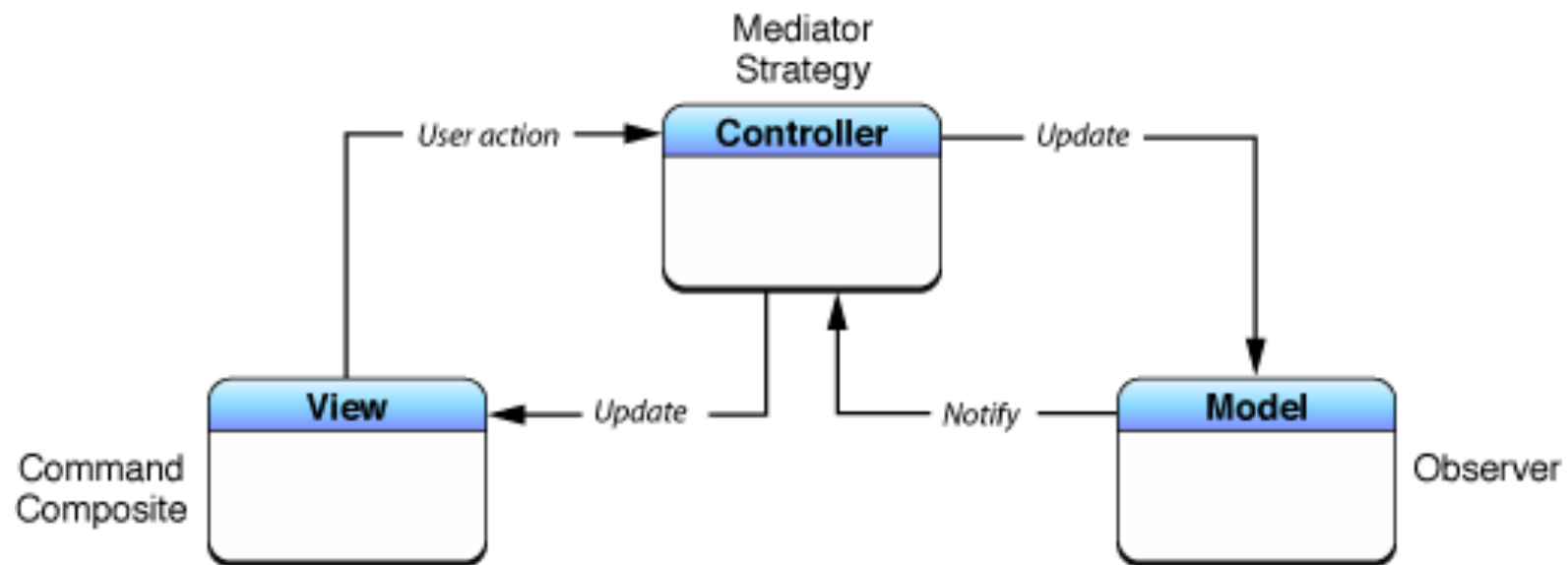
Method selector → setOriginX:y:

Delegation Pattern



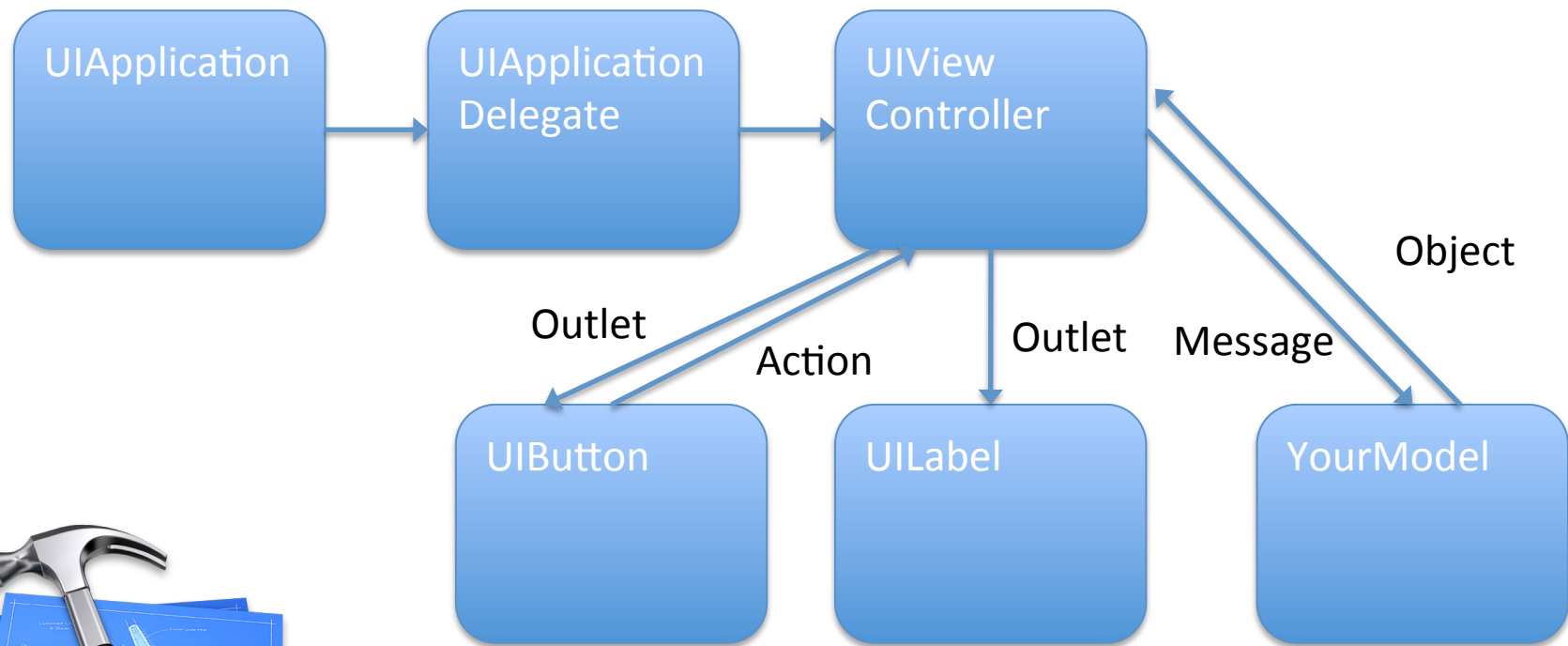
Apple Developer Library, "Cocoa Design Patterns"

Model View Controller Pattern



Apple Developer Library, "Cocoa Design Patterns"

View Based Application: Simplified



Build a View Based Application from Project Templates

Assignment

- Read in Apple Developer Library:
 - Xcode 4 User Guide
 - Learning Objective-C a primer
 - The Objective-C Programming Language
- Read Kochan:
 - Chapters 1-3, 21 and Glossary
- Complete Reading/Participation assignments by Thursday, Sept. 1st.