Design Patterns

MSc in Computer Science

Produced by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology

http://www.wit.ie

http://elearning.wit.ie

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

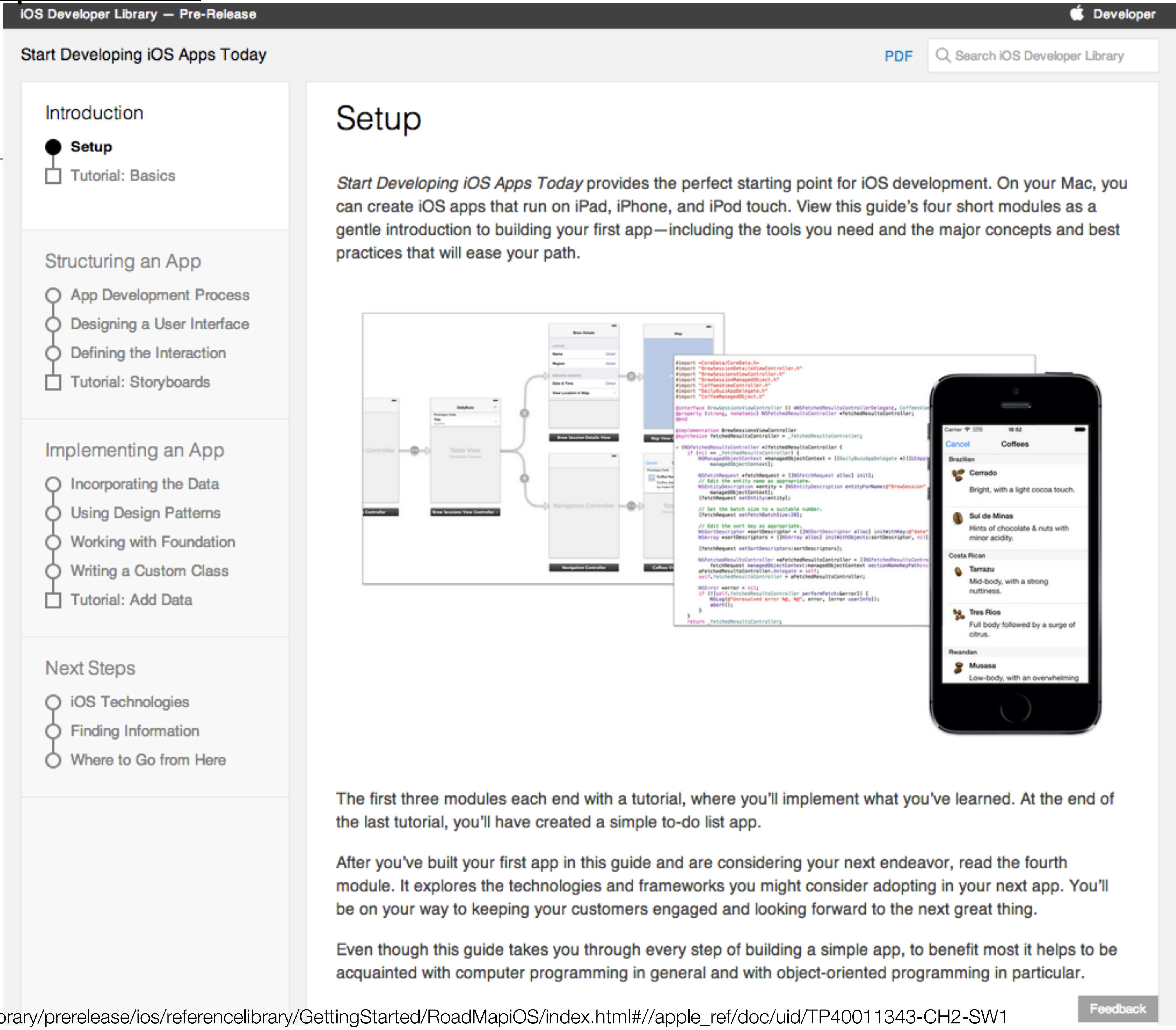
eLearning support unit

Todo App

developer.apple.com

Start Developing iOS Apps Today

Objective-C only



iOS Developer Library — Pre-Release

Developer

Start Developing iOS Apps Today

PDF

Search iOS Developer Library

Setup

Start Developing iOS Apps Today provides the perfect starting point for iOS development. On your Mac, you can create iOS apps that run on iPad, iPhone, and iPod touch. View this guide's four short modules as a gentle introduction to building your first app—including the tools you need and the major concepts and best practices that will ease your path.

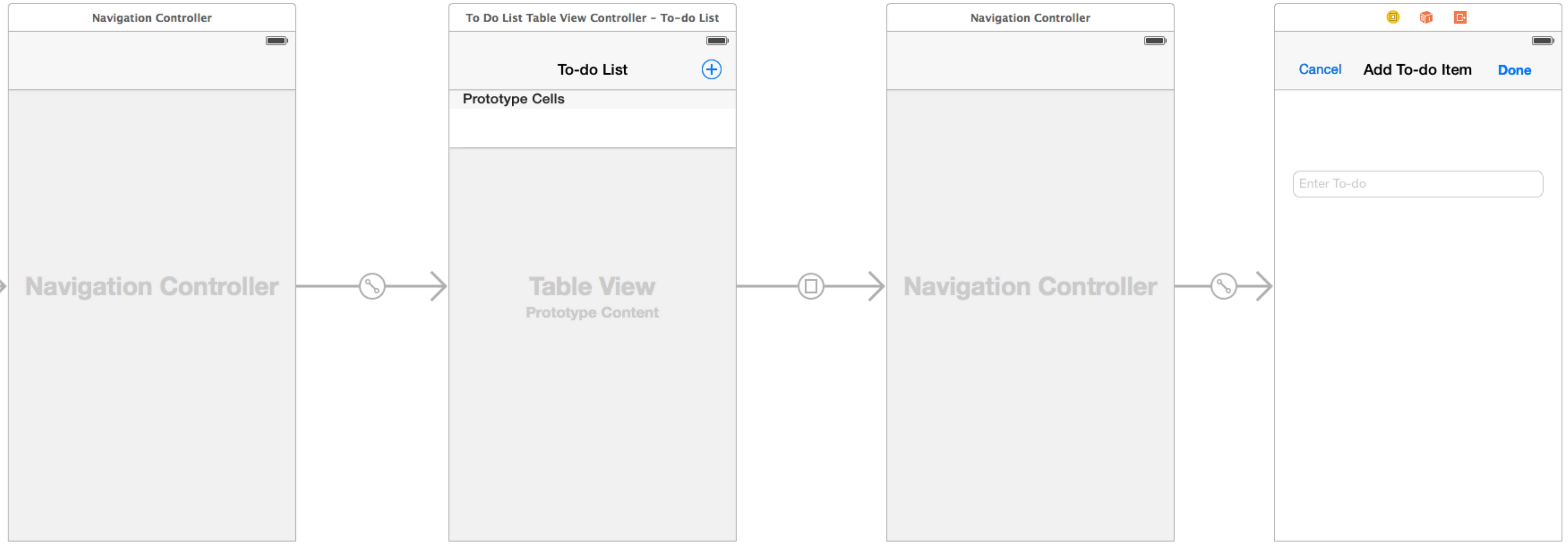
The first three modules each end with a tutorial, where you'll implement what you've learned. At the end of the last tutorial, you'll have created a simple to-do list app.

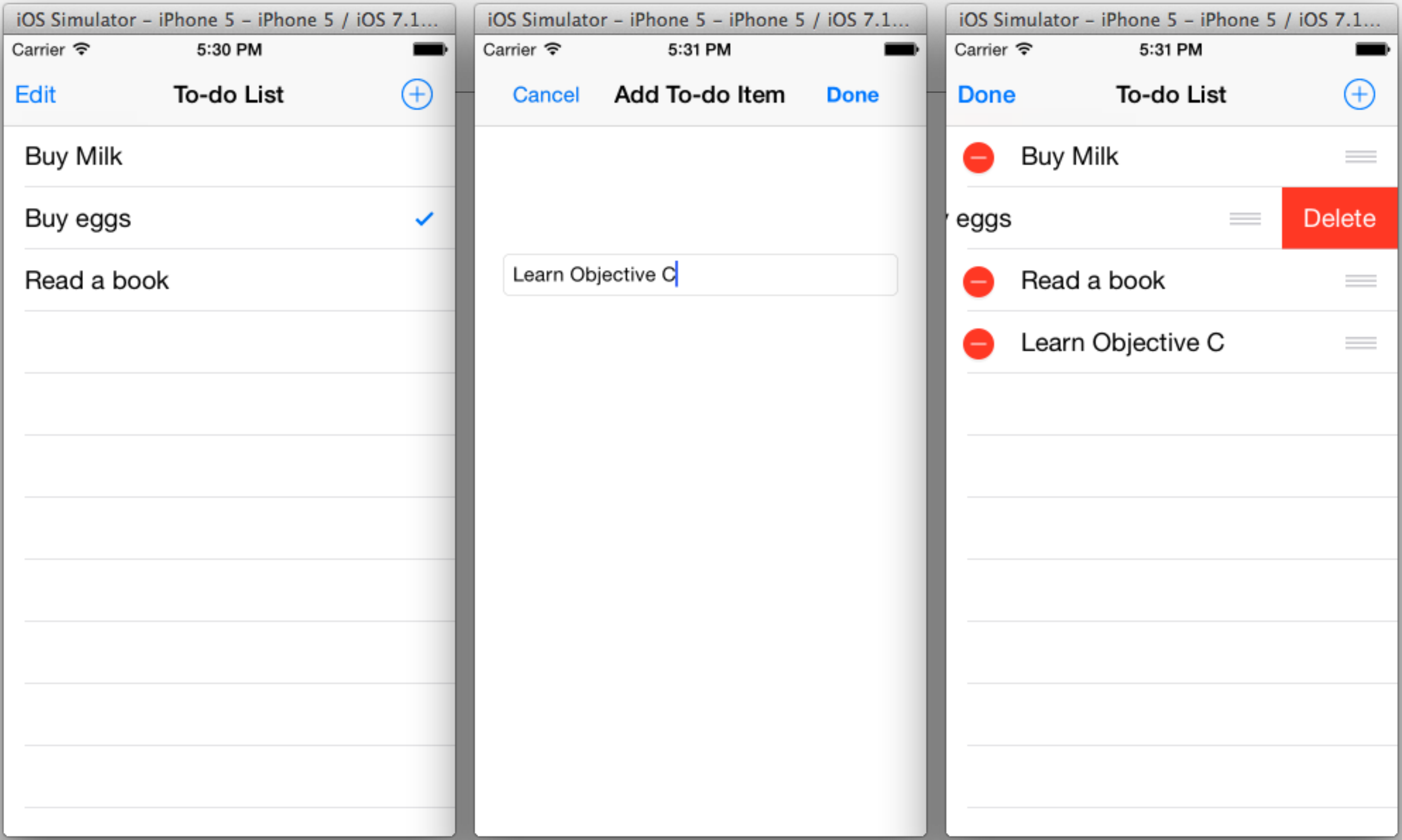
After you've built your first app in this guide and are considering your next endeavor, read the fourth module. It explores the technologies and frameworks you might consider adopting in your next app. You'll be on your way to keeping your customers engaged and looking forward to the next great thing.

Even though this guide takes you through every step of building a simple app, to benefit most it helps to be acquainted with computer programming in general and with object-oriented programming in particular.

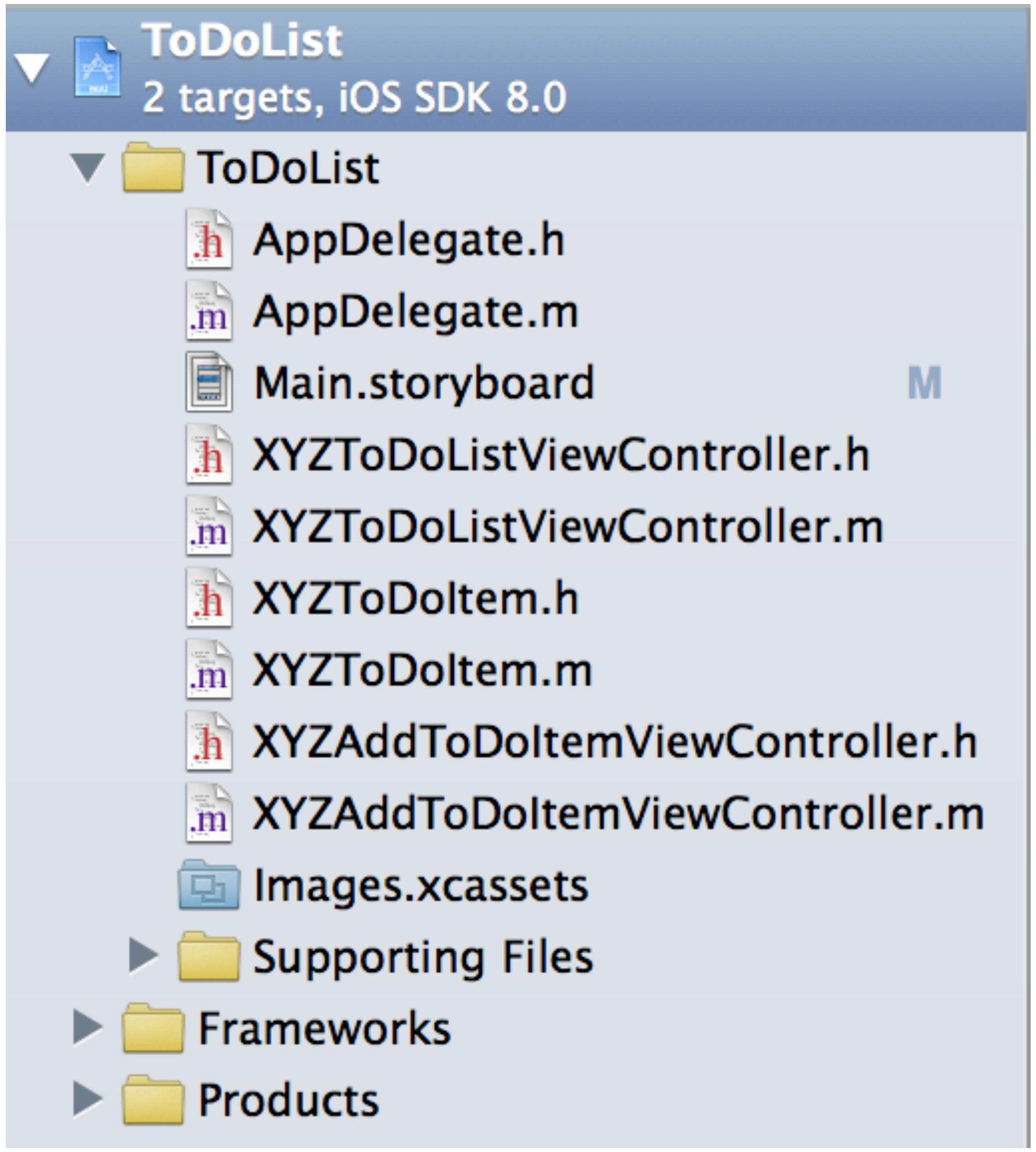
Feedback

https://developer.apple.com/library/prerelease/ios/referencelibrary/GettingStarted/RoadMapiOS/index.html#//apple_ref/doc/uid/TP40011343-CH2-SW1


Navigation Controller
Navigation Controller
To Do List Table View Controller – To-do List
To-do List
Prototype Cells
Table View
Prototype Content
Navigation Controller
Navigation Controller
Cancel
Add To-do Item
Done
Enter To-do

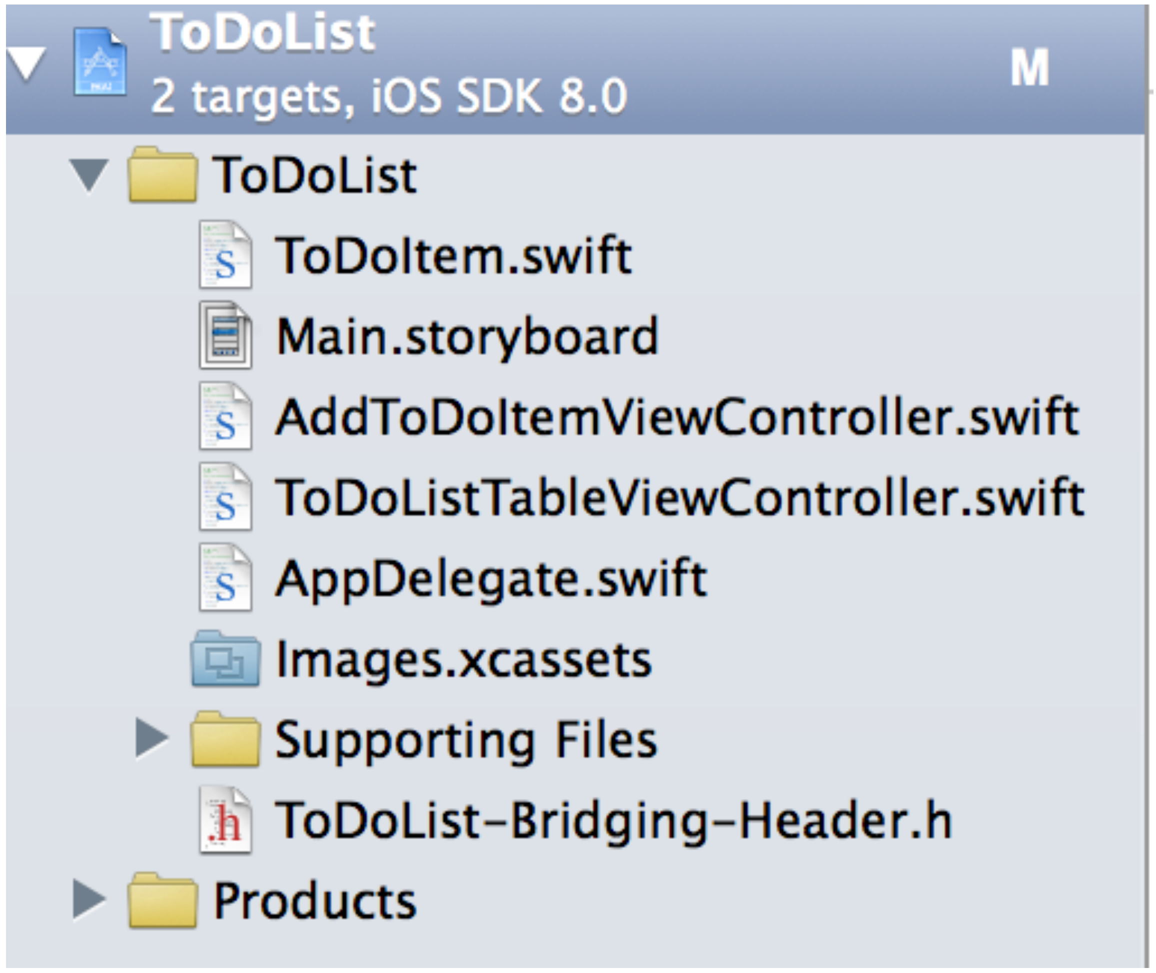

iOS Simulator – iPhone 5 – iPhone 5 / iOS 7.1…
Carrier
5:30 PM
Edit
To-do List
Buy Milk
Buy eggs
Read a book
iOS Simulator – iPhone 5 – iPhone 5 / iOS 7.1…
Carrier
5:31 PM
Cancel
Add To-do Item
Done
Learn Objective C
iOS Simulator – iPhone 5 – iPhone 5 / iOS 7.1…
Carrier
5:31 PM
Done
To-do List
Buy Milk
eggs
Delete
Read a book
Learn Objective C

ToDoList Applications



```
ToDoList
2 targets, iOS SDK 8.0
  ToDoList
    AppDelegate.h
    AppDelegate.m
    Main.storyboard                    M
    XYZToDoListViewController.h
    XYZToDoListViewController.m
    XYZToDoItem.h
    XYZToDoItem.m
    XYZAddToDoItemViewController.h
    XYZAddToDoItemViewController.m
    Images.xcassets
    Supporting Files
  Frameworks
  Products
```

8 source files



```
ToDoList                               M
2 targets, iOS SDK 8.0
  ToDoList
    ToDoItem.swift
    Main.storyboard
    AddToDoItemViewController.swift
    ToDoListTableViewController.swift
    AppDelegate.swift
    Images.xcassets
    Supporting Files
    ToDoList-Bridging-Header.h
  Products
```

5 source files

AppDelegate

```
#import <UIKit/UIKit.h>

@interface AppDelegate : UIResponder <UIApplicationDelegate>

  @property (strong, nonatomic) UIWindow *window;

@end
```

```
#import "AppDelegate.h"

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application
-       didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
  return YES;
}

- (void)applicationWillResignActive:(UIApplication *)application
{
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
}

- (void)applicationWillEnterForeground:(UIApplication *)application
{
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
}

- (void)applicationWillTerminate:(UIApplication *)application
{
}

@end
```

```
import UIKit

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate
{
  var window: UIWindow?

  func application(application: UIApplication,
                   didFinishLaunchingWithOptions: NSDictionary?) -> Boo
  {
    return true
  }

  func applicationWillResignActive(application: UIApplication)
  {
  }

  func applicationDidEnterBackground(application: UIApplication)
  {
  }

  func applicationWillEnterForeground(application: UIApplication)
  {
  }

  func applicationDidBecomeActive(application: UIApplication)
  {
  }

  func applicationWillTerminate(application: UIApplication)
  {
  }
}
```

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

@interface XYZToDoItem : NSObject <NSCoding>

  @property NSString *itemName;
  @property BOOL      completed;

@end
```

The Model

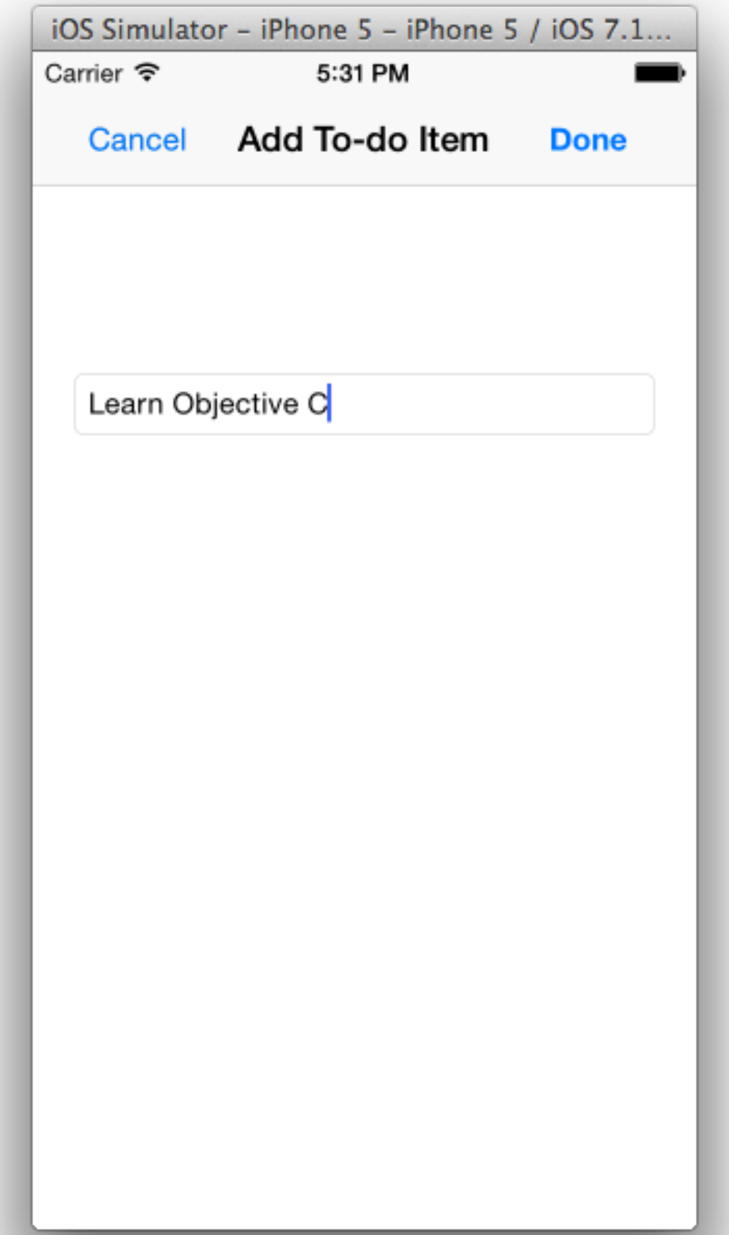
```objc
#import "XYZToDoItem.h"

@implementation XYZToDoItem
- (id)init
{
  self = [super init];
  if (self)
  {
    _completed = NO;
  }

  return self;
}

- (id)initWithCoder:(NSCoder *)coder
{
  self = [super init];
  if (self)
  {
    _itemName = [coder decodeObjectForKey:@"itemName"];
    _completed = [coder decodeBoolForKey:@"completed"];
  }

  return self;
}

- (void)encodeWithCoder:(NSCoder *)coder
{
  [coder encodeObject:self.itemName forKey:@"itemName"];
  [coder encodeBool:self.completed forKey:@"completed"];
}

@end
```

```swift
class ToDoItem
{
  var completed = false
  var itemName  = ""

  init(completed: Bool = false, itemName:String = "empty")
  {
    self.completed = completed
    self.itemName  = itemName
  }
}
```

ToDoItemController

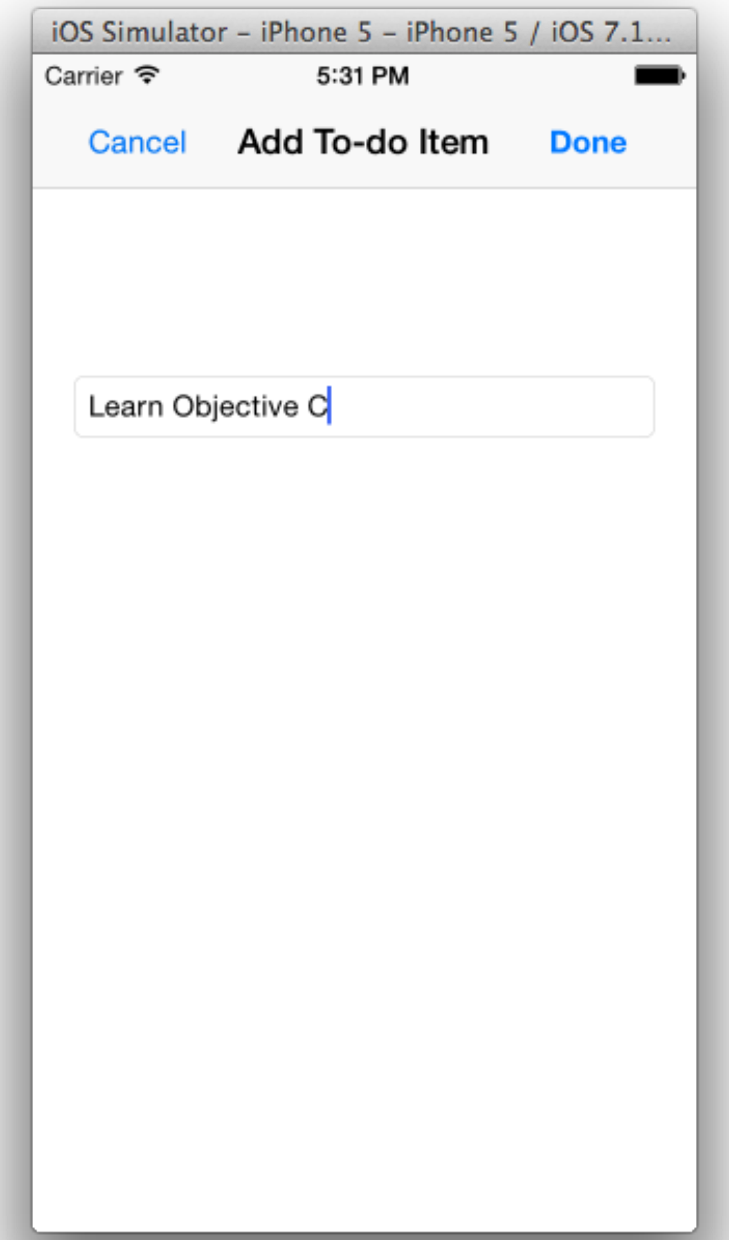
```
#import <UIKit/UIKit.h>
#import "XYZToDoItem.h"

@interface XYZAddToDoItemViewController : UIViewController
  @property XYZToDoItem *toDoItem;
@end
```

```
#import "XYZAddToDoItemViewController.h"

@interface XYZAddToDoItemViewController ()
  @property (weak, nonatomic) IBOutlet UITextField *textField;
  @property (weak, nonatomic) IBOutlet UIBarButtonItem *doneButton;
@end

@implementation XYZAddToDoItemViewController

- (void) prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
  if (sender != self.doneButton) return;
  if (self.textField.text.length > 0)
  {
    self.toDoItem = [[XYZToDoItem alloc] init];
    self.toDoItem.itemName = self.textField.text;
  }
}

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
  self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
  if (self)
  {
  }
  return self;
}

- (void)viewDidLoad
{
  [super viewDidLoad];
}

- (void)didReceiveMemoryWarning
{
  [super didReceiveMemoryWarning];
}
@end
```



ToDoItemController

```
import UIKit

class AddToDoItemViewController: UIViewController
{
  var todoItem : ToDoItem?

  @IBOutlet var toDoItemText : UITextField
  @IBOutlet var doneButton   : UIButton

  init(nibName nibNameOrNil: String?, bundle nibBundleOrNil: NSBundle?)
  {
    super.init(nibName: nibNameOrNil, bundle: nibBundleOrNil)
  }

  override func prepareForSegue(segue: UIStoryboardSegue!, sender: AnyObject!)
  {
    if let button = sender as? NSObject
    {
      todoItem = button == doneButton ? ToDoItem(itemName:toDoItemText.text): nil
    }
  }

  init(coder aDecoder: NSCoder!)
  {
    super.init(coder: aDecoder)
  }
}
```



ToDoListTableConroller

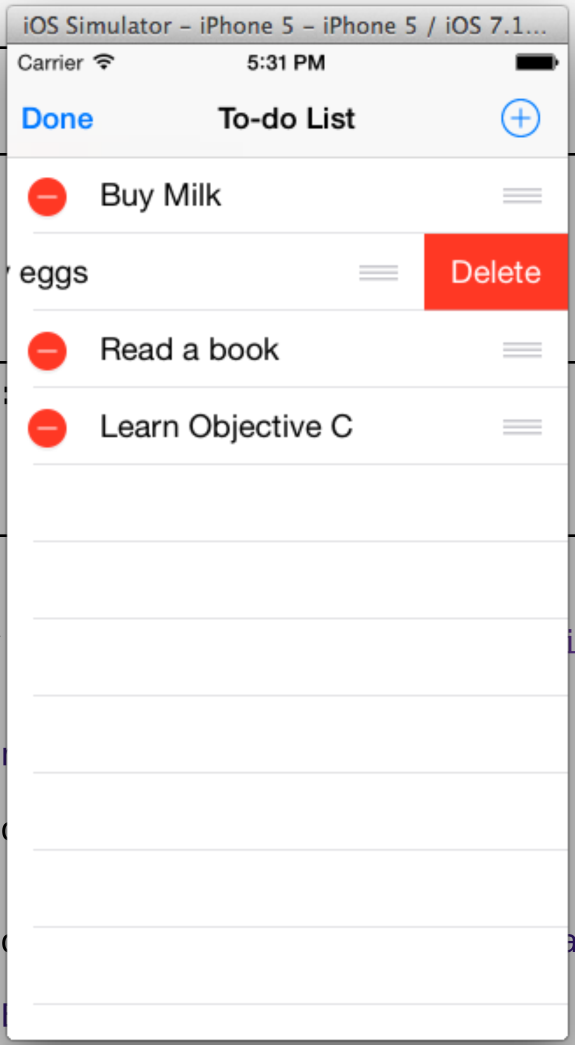
```
#import <UIKit/UIKit.h>

@interface XYZToDoListViewController : UITableViewController
  @property NSMutableArray *toDoItems;
@end
```

```
#import "XYZToDoListViewController.h"
#import "XYZToDoItem.h"
#import "XYZAddToDoItemViewController.h"

@interface XYZToD

@property NSStrin

@end

@implementation X

- (void)saveList
{
  [NSKeyedArchive
}

- (void)loadInit
{
  XYZToDoItem *it
  item1.itemName
  [self.toDoItems

  XYZToDoItem *it
  item2.itemName
  [self.toDoItems

  XYZToDoItem *it
  item3.itemName
  [self.toDoItems

}

- (IBAction)unwin
{
  XYZAddToDoItemV
  XYZToDoItem *it
  if (item != nil
  {
    [self.toDoIte
    [self saveLis
    [self.tableVi

  }
}

  (id)initWithSty
```

```
- (id)initWithStyle:(UITableViewStyle)style
{
  self = [super initWithStyle:style];
  if (self)
  {
  }
  return self;
}

- (void)viewDidLoad
{
  [super viewDidLoad]

  self.path = [NSSear
  self.path = [self.pa

  NSFileManager *fileM
  if (![fileManager f
  {
    self.toDoItems =
  }
  else
  {
    self.toDoItems =
  }
  self.navigationItem
}

- (void)didReceiveMem
{
  [super didReceiveMem
}

#pragma mark - Table

- (NSInteger)numberOf
{
  return 1;
}

- (NSInteger)tableVie
{
  return [self.toDoIt
}
```

```
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:
{
  return [self.toDoItems count];
}

- (UITableViewCell *)tableV
{
  static NSString *CellIder
  UITableViewCell *cell =

  XYZToDoItem *toDoItem =
  cell.textLabel.text = toD

  if (toDoItem.completed)
  {
    cell.accessoryType = UI
  }
  else
  {
    cell.accessoryType = UI
  }

  return cell;
}


- (BOOL)tableView:(UITableV
{
  return YES;
}

- (void)tableView:(UITableV
*)indexPath
{
  if (editingStyle == UITab
  {
    [self.toDoItems remove
    [self saveList];
```

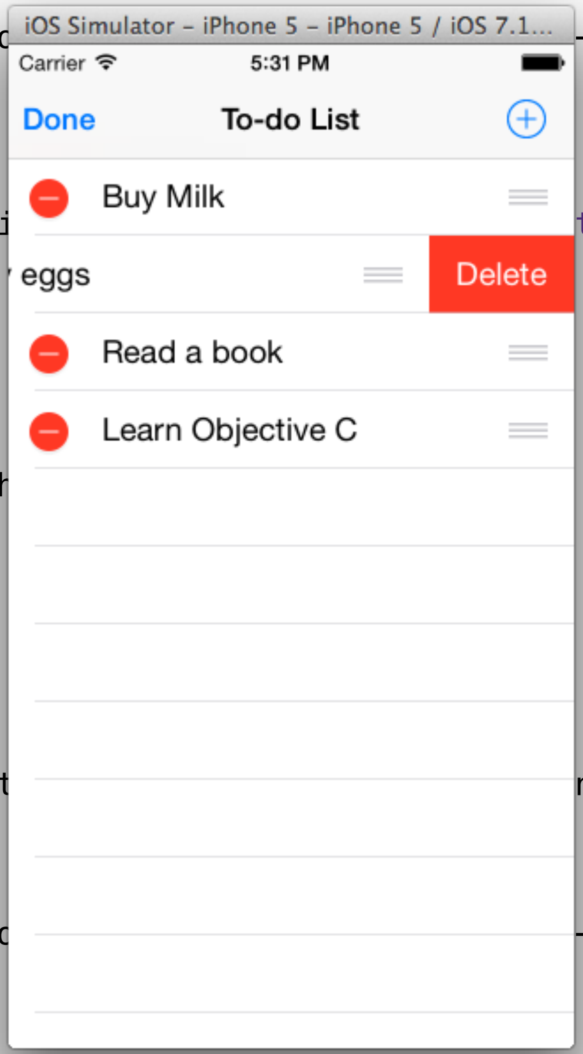
```
- (void)tableView:(UITableView *)tableView                       iew
*)indexPath
{
  if (editingStyle == UITableViewCellEditi
  {
    [self.toDoItems removeObjectAtIndex:in
    [self saveList];

    [tableView deleteRowsAtIndexPaths:@[in                           abl
  }
  else if (editingStyle == UITableViewCellI
  {
  }
}

- (void)tableView:(UITableView *)tableView moveRowAtIndexPath:(NSIndexPath
{
}

- (BOOL)tableView:(UITableView *)tableView canMoveRowAtIndexPath:(NSIndexP
{
  return YES;
}


#pragma mark - Navigation

- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSInde
{
  [tableView deselectRowAtIndexPath:indexPath animated:NO];
```



ToDoListTableConroller

```
import UIKit

@objc(ToDoListTableViewController) class ToDoListTableViewController: UITableViewController
{
  var todoItems = ToDoItem[]()

  init(style: UITableViewStyle)
  {
    super.init(style: style)
  }

  init(coder aDecoder: NSCoder!)
  {
    super.init(coder: aDecoder)
  }

  func loadInitialData()
  {
    todoItems.append(ToDoItem(itemName:"Buy Mil
    todoItems.append(ToDoItem(itemName:"Buy egg
    todoItems.append(ToDoItem(itemName:"Read a
  }

  override func viewDidLoad()
  {
    super.viewDidLoad()
    loadInitialData()
    navigationItem.leftBarButtonItem = self.edi
  }

  @IBAction func unwindToList (segue: UIStorybo
  {
    var controller = segue?.sourceViewControlle
    if controller.todoItem != nil
    {
      todoItems.append(controller.todoItem!)
      self.tableView.reloadData()
    }
  }

  override func numberOfSectionsInTableView(tab
  {
    return 1
  }

  override func tableView(tableView: UITableVie
  {
    return todoItems.count
  }
```

```
  override func tableView(tableView: UITableView!, didSelectRowAtIndexPath : NSIndexPath!)
  {
    tableView.deselectRowAtIndexPath(didSelectRowAtIndexPath, animated: false)
    var task = self.todoItems[didSelectRowAtIndexPath.row] as ToDoItem
    task.completed = !task.completed
    tableView.reloadRowsAtIndexPaths([didSelectRowAtIndexPath], withRowAnimation: UITableVie
  }

  override func tableView(tableView: UITableView?, cellForRowAtIndexPath : NSIndexPath!) ->
  {
    let cell           = UITableViewCell(style: UITableViewCellStyle.Default, reuseIdentifie
    var task           = todoItems[cellForRowAtIndexPath.row]

    cell.text          = task.itemName
    cell.accessoryType = task.completed ? UITableViewCellAccessoryType.Checkmark : UITableVi

    return cell
  }

  override func tableView(tableView: UITableView?, canEditRo                        ->
  {
    return true
  }

  override func tableView(tableView: UITableView?, commitEdi                        tin
  {
    if commitEditingStyle == .Delete
    {
      if let index = forRowAtIndexPath?.row
      {
        todoItems.removeAtIndex(index)
        tableView?.deleteRowsAtIndexPaths([forRowAtIndexPath
      }
    }
    else if commitEditingStyle == .Insert
    {
    }
  }

  override func tableView(tableView: UITableView?, moveRowAt                        nde
  {
  }

  override func tableView(tableView: UITableView?, canMoveRo                        ->
  {
    return true
  }
}
```



KLocs

Objctive-C		**Swift**	
AppDelegate.h	7		
AppDelegate.m	30	AppDelegate.swift	33
ToDoItem.h	8		
ToDoItem.m	33	ToDoItem.swift	12
ToDoItemController.h	8		
ToDoItemController.m	41	ToDoItemController.swift	27
ToDiListController.h	7		
ToDiListController.m	156	ToDoListController.swift	98
	290		**170**