```
//
//  Account.h
//  ATM
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Simulates an ATM (that only accepts deposits).
//

#import <Foundation/Foundation.h>

@interface Account : NSObject {
}

@property (assign, nonatomic) unsigned long long balance;

@end
```

```
//
//  Account.m
//  ATM
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Simulates an ATM (that only accepts deposits).
//

#import "Account.h"

@implementation Account

@synthesize balance=_balance;

- (id)init
{
    if (self = [super init]) {
        self.balance = 0;
    }
    return self;
}

@end
```

```
//
//  AppDelegate.h
//  ATM
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Simulates an ATM (that only accepts deposits).
//

#import <UIKit/UIKit.h>

@class ViewController;

@interface AppDelegate : UIResponder <UIApplicationDelegate>

@property (strong, nonatomic) ViewController *viewController;
@property (strong, nonatomic) UIWindow *window;

@end
```

```
//
//  AppDelegate.m
//  ATM
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Simulates an ATM (that only accepts deposits).
//

#import "AppDelegate.h"

#import "ViewController.h"

@implementation AppDelegate

@synthesize viewController = _viewController;
@synthesize window = _window;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    self.viewController = [[ViewController alloc] initWithNibName:@"ViewController" bundle:nil];
    self.window.rootViewController = self.viewController;
    [self.window makeKeyAndVisible];
    return YES;
}

@end
```

```
//
//  ViewController.h
//  ATM
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Simulates an ATM (that only accepts deposits).
//

#import <UIKit/UIKit.h>
#import "Account.h"

@interface ViewController : UIViewController {
}

@property (nonatomic, strong) Account *account;
@property (assign, nonatomic) unsigned long long amount;
@property (nonatomic, weak) IBOutlet UILabel *balanceLabel;
@property (nonatomic, weak) IBOutlet UILabel *depositLabel;

- (IBAction)clear:(id)sender;
- (IBAction)deposit:(id)sender;
- (IBAction)digit:(id)sender;
- (void)show;

@end
```

```
1.  //
2.  //  ViewController.m
3.  //  ATM
4.  //
5.  //  David J. Malan
6.  //  Harvard University
7.  //  malan@harvard.edu
8.  //
9.  //  Simulates an ATM (that only accepts deposits).
10. //
11.
12. #import "ViewController.h"
13.
14. @implementation ViewController
15.
16. @synthesize account=_account;
17. @synthesize amount=_amount;
18. @synthesize balanceLabel=_balanceLabel;
19. @synthesize depositLabel=_depositLabel;
20.
21. - (IBAction)clear:(id)sender
22. {
23.     // clear input
24.     self.amount = 0;
25.     [self show];
26. }
27.
28. - (IBAction)deposit:(id)sender
29. {
30.     // deposit amount
31.     self.account.balance += self.amount;
32.
33.     // clear input
34.     self.amount = 0;
35.     [self show];
36. }
37.
38. - (IBAction)digit:(id)sender
39. {
40.     // append digit to amount
41.     UIButton *b = (UIButton *)sender;
42.     self.amount = self.amount * 10 + b.tag;
43.     [self show];
44. }
45.
46. - (void)show
47. {
48.     // show balance
```

```
49.     self.balanceLabel.text = [NSString stringWithFormat:@"$%llu", self.account.balance];
50.
51.     // show input
52.     self.depositLabel.text = [NSString stringWithFormat:@"$%llu", self.amount];
53. }
54.
55. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
56. {
57.     return (interfaceOrientation == UIInterfaceOrientationPortrait);
58. }
59.
60. - (void)viewDidLoad
61. {
62.     // create account
63.     self.account = [[Account alloc] init];
64.     [self show];
65. }
66.
67. @end
```

```
//
//  ATMApplicationTests.h
//  ATMApplicationTests
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Simulates an ATM (that only accepts deposits).
//

#import <SenTestingKit/SenTestingKit.h>

@interface ATMApplicationTests : SenTestCase
@end
```

```
1.  //
2.  //  ATMApplicationTests.m
3.  //  ATMApplicationTests
4.  //
5.  //  David J. Malan
6.  //  Harvard University
7.  //  malan@harvard.edu
8.  //
9.  //  Simulates an ATM (that only accepts deposits).
10. //
11.
12. #import "ATMApplicationTests.h"
13.
14. #import "AppDelegate.h"
15. #import "ViewController.h"
16.
17.
18. @interface ATMApplicationTests ()
19.
20. // private properties
21. @property (nonatomic, readwrite, weak) AppDelegate *appDelegate;
22. @property (nonatomic, readwrite, weak) ViewController *viewController;
23. @property (nonatomic, readwrite, weak) UIView *view;
24.
25. @end
26.
27.
28. @implementation ATMApplicationTests
29.
30. @synthesize appDelegate=_appDelegate;
31. @synthesize viewController=_viewController;
32. @synthesize view=_view;
33.
34. - (void)setUp
35. {
36.     [super setUp];
37.
38.     self.appDelegate = [[UIApplication sharedApplication] delegate];
39.     self.viewController = self.appDelegate.viewController;
40.     self.view = self.viewController.view;
41. }
42.
43. - (void)testAppDelegate
44. {
45.     STAssertNotNil(self.appDelegate, @"Cannot find the application delegate");
46. }
47.
48. - (void)testClear
```

```
49. {
50.     // input $1
51.     [self.viewController digit:[self.view viewWithTag:1]];
52.     STAssertTrue([[self.viewController.depositLabel text] isEqualToString:@"$1"], @"Deposit should be $1");
53. 
54.     // clearing should yield $0
55.     [self.viewController clear:nil];
56.     STAssertTrue([[self.viewController.depositLabel text] isEqualToString:@"$0"], @"Deposit should be $0");
57. }
58. 
59. - (void)testDeposit
60. {
61.     // balance should be $0 at first
62.     STAssertTrue([[self.viewController.balanceLabel text] isEqualToString:@"$0"], @"Balance should be $0");
63. 
64.     // deposit $1
65.     [self.viewController digit:[self.view viewWithTag:1]];
66.     STAssertTrue([[self.viewController.depositLabel text] isEqualToString:@"$1"], @"Deposit should be $1");
67.     [self.viewController deposit:nil];
68.     STAssertTrue([[self.viewController.depositLabel text] isEqualToString:@"$0"], @"Deposit should be $0");
69.     STAssertTrue([[self.viewController.balanceLabel text] isEqualToString:@"$1"], @"Balance should be $1");
70. 
71.     // deposit $12
72.     [self.viewController digit:[self.view viewWithTag:1]];
73.     [self.viewController digit:[self.view viewWithTag:2]];
74.     STAssertTrue([[self.viewController.depositLabel text] isEqualToString:@"$12"], @"Deposit should be $12");
75.     [self.viewController deposit:nil];
76.     STAssertTrue([[self.viewController.depositLabel text] isEqualToString:@"$0"], @"Deposit should be $0");
77.     STAssertTrue([[self.viewController.balanceLabel text] isEqualToString:@"$13"], @"Balance should be $13");
78. }
79. 
80. - (void)testDigit
81. {
82.     // default value should be $0
83.     STAssertTrue([[self.viewController.depositLabel text] isEqualToString:@"$0"], @"Deposit should be $0");
84. 
85.     // inputting a leading zero should still yield $0
86.     [self.viewController digit:[self.view viewWithTag:0]];
87.     STAssertTrue([[self.viewController.depositLabel text] isEqualToString:@"$0"], @"Deposit should be $0");
88. 
89.     // inputting 1 should yield $1
90.     [self.viewController digit:[self.view viewWithTag:1]];
91.     STAssertTrue([[self.viewController.depositLabel text] isEqualToString:@"$1"], @"Deposit should be $1");
92. 
93.     // inputting 2 should now yield $12
94.     [self.viewController digit:[self.view viewWithTag:2]];
95.     STAssertTrue([[self.viewController.depositLabel text] isEqualToString:@"$12"], @"Deposit should be $12");
96. }
```

```
 97.
 98. - (void)tearDown
 99. {
100.     // Tear-down code here.
101.
102.     [super tearDown];
103. }
104.
105. - (void)testExample
106. {
107.     //STFail(@"Unit tests are not implemented yet in ATMApplicationTests");
108. }
109.
110. @end
```

```
//
//  ATMLogicTests.h
//  ATMLogicTests
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Simulates an ATM (that only accepts deposits).
//

#import <SenTestingKit/SenTestingKit.h>

@interface ATMLogicTests : SenTestCase
@end
```

```
//
//  ATMLogicTests.m
//  ATMLogicTests
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Simulates an ATM (that only accepts deposits).
//

#import "ATMLogicTests.h"
#import "Account.h"


@interface ATMLogicTests ()

// private property
@property (nonatomic, readwrite, strong) Account *account;

@end


@implementation ATMLogicTests

@synthesize account=_account;

- (void)setUp
{
    [super setUp];

    // Set-up code here.
    NSLog(@"setUp");
    self.account = [[Account alloc] init];
    STAssertNotNil(self.account, @"Cannot create Account instance");
}

- (void)testBalance
{
    // balance should be $0 initially
    STAssertTrue(self.account.balance == 0, @"Balance is not 0");
}

- (void)tearDown
{
    [super tearDown];

    // Tear-down code here.
```

```
49.     NSLog(@"tearDown");
50. }
51.
52. @end
```

```
//
//  AppDelegate.h
//  Hola1
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates localized nibs.
//

#import <UIKit/UIKit.h>

@class ViewController;

@interface AppDelegate : UIResponder <UIApplicationDelegate>

@property (strong, nonatomic) ViewController *viewController;
@property (strong, nonatomic) UIWindow *window;

@end
```

```
//
//  AppDelegate.m
//  Hola1
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates localized nibs.
//

#import "AppDelegate.h"
#import "ViewController.h"

@implementation AppDelegate

@synthesize viewController=_viewController;
@synthesize window=_window;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    self.viewController = [[ViewController alloc] initWithNibName:@"ViewController" bundle:nil];
    self.window.rootViewController = self.viewController;
    [self.window makeKeyAndVisible];
    return YES;
}

@end
```

```
//
//  ViewController.h
//  Hola1
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates localized nibs.
//

#import <UIKit/UIKit.h>

@interface ViewController : UIViewController
@end
```

```
//
//  ViewController.m
//  Hola1
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates localized nibs.
//

#import "ViewController.h"

@implementation ViewController

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
{
    return YES;
}

@end
```

```h
//
//  AppDelegate.h
//  Hola2
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates localized strings.
//

#import <UIKit/UIKit.h>

@class ViewController;

@interface AppDelegate : UIResponder <UIApplicationDelegate>

@property (strong, nonatomic) ViewController *viewController;
@property (strong, nonatomic) UIWindow *window;

@end
```

```
//
//  AppDelegate.m
//  Hola2
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates localized strings.
//

#import "AppDelegate.h"

#import "ViewController.h"

@implementation AppDelegate

@synthesize viewController=_viewController;
@synthesize window=_window;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    self.viewController = [[ViewController alloc] initWithNibName:@"ViewController" bundle:nil];
    self.window.rootViewController = self.viewController;
    [self.window makeKeyAndVisible];
    return YES;
}

@end
```

```
//
//  ViewController.h
//  Hola2
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates localized strings.
//

#import <UIKit/UIKit.h>

@interface ViewController : UIViewController
@end
```

```
//
//  ViewController.m
//  Hola2
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates localized strings.
//

#import "ViewController.h"


@interface ViewController ()
@property (nonatomic, readwrite, weak) IBOutlet UILabel *label;
@end


@implementation ViewController

@synthesize label=_label;

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
{
    return YES;
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    // localize greeting
    self.label.text = NSLocalizedString(@"GREETING", nil);
}

@end
```

```
/*
     File: Calculator.h
 Abstract: This file declares the interface for the Calculator class.
  Version: 2.0

 Disclaimer: IMPORTANT:  This Apple software is supplied to you by Apple
 Inc. ("Apple") in consideration of your agreement to the following
 terms, and your use, installation, modification or redistribution of
 this Apple software constitutes acceptance of these terms.  If you do
 not agree with these terms, please do not use, install, modify or
 redistribute this Apple software.

 In consideration of your agreement to abide by the following terms, and
 subject to these terms, Apple grants you a personal, non-exclusive
 license, under Apple's copyrights in this original Apple software (the
 "Apple Software"), to use, reproduce, modify and redistribute the Apple
 Software, with or without modifications, in source and/or binary forms;
 provided that if you redistribute the Apple Software in its entirety and
 without modifications, you must retain this notice and the following
 text and disclaimers in all such redistributions of the Apple Software.
 Neither the name, trademarks, service marks or logos of Apple Inc. may
 be used to endorse or promote products derived from the Apple Software
 without specific prior written permission from Apple.  Except as
 expressly stated in this notice, no other rights or licenses, express or
 implied, are granted by Apple herein, including but not limited to any
 patent rights that may be infringed by your derivative works or by other
 works in which the Apple Software may be incorporated.

 The Apple Software is provided by Apple on an "AS IS" basis.  APPLE
 MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION
 THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS
 FOR A PARTICULAR PURPOSE, REGARDING THE APPLE SOFTWARE OR ITS USE AND
 OPERATION ALONE OR IN COMBINATION WITH YOUR PRODUCTS.

 IN NO EVENT SHALL APPLE BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL
 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 INTERRUPTION) ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION,
 MODIFICATION AND/OR DISTRIBUTION OF THE APPLE SOFTWARE, HOWEVER CAUSED
 AND WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE),
 STRICT LIABILITY OR OTHERWISE, EVEN IF APPLE HAS BEEN ADVISED OF THE
 POSSIBILITY OF SUCH DAMAGE.

 Copyright (C) 2011 Apple Inc. All Rights Reserved.

 */

/*!
```

```
 * @class Calculator
 * This class implements a key-press--based calculator engine.
 * @throws NSInvalidArgumentException
 * @updated 2009-04-22
 */

#import <Foundation/Foundation.h>

@interface Calculator : NSObject {
@private
    NSMutableString *_display;    // The calculator display (the value a harwdare-based calculator shows on its LCD screen).
    double           _operand;
    NSString        *_operator;
}

- init;
- (void) dealloc;


/*!
 * @method input:
 * Receives input into the calculator.
 *
 * Valid characters:
 *
 *     Digits:    .0123456789
 *
 *     Operators: +-Ã—/=
 *
 *     Commands:  D     Delete
 *                C     Clear
 *
 * @throws NSInvalidArgumentException when character is not valid.
 */
- (void) input:(NSString *) character;


/*!
 * @method displayValue
 * Provides the value in the calculatorâ€™s â€œdisplay.â€•
 */
- (NSString *) displayValue;

@end
```

```
/*
     File: Calculator.m
 Abstract: This file implements the Calculator class.
  Version: 2.0

 Disclaimer: IMPORTANT:  This Apple software is supplied to you by Apple
 Inc. ("Apple") in consideration of your agreement to the following
 terms, and your use, installation, modification or redistribution of
 this Apple software constitutes acceptance of these terms.  If you do
 not agree with these terms, please do not use, install, modify or
 redistribute this Apple software.

 In consideration of your agreement to abide by the following terms, and
 subject to these terms, Apple grants you a personal, non-exclusive
 license, under Apple's copyrights in this original Apple software (the
 "Apple Software"), to use, reproduce, modify and redistribute the Apple
 Software, with or without modifications, in source and/or binary forms;
 provided that if you redistribute the Apple Software in its entirety and
 without modifications, you must retain this notice and the following
 text and disclaimers in all such redistributions of the Apple Software.
 Neither the name, trademarks, service marks or logos of Apple Inc. may
 be used to endorse or promote products derived from the Apple Software
 without specific prior written permission from Apple.  Except as
 expressly stated in this notice, no other rights or licenses, express or
 implied, are granted by Apple herein, including but not limited to any
 patent rights that may be infringed by your derivative works or by other
 works in which the Apple Software may be incorporated.

 The Apple Software is provided by Apple on an "AS IS" basis.  APPLE
 MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION
 THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS
 FOR A PARTICULAR PURPOSE, REGARDING THE APPLE SOFTWARE OR ITS USE AND
 OPERATION ALONE OR IN COMBINATION WITH YOUR PRODUCTS.

 IN NO EVENT SHALL APPLE BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL
 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 INTERRUPTION) ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION,
 MODIFICATION AND/OR DISTRIBUTION OF THE APPLE SOFTWARE, HOWEVER CAUSED
 AND WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE),
 STRICT LIABILITY OR OTHERWISE, EVEN IF APPLE HAS BEEN ADVISED OF THE
 POSSIBILITY OF SUCH DAMAGE.

 Copyright (C) 2011 Apple Inc. All Rights Reserved.

 */


```

```
49. #import "Calculator.h"
50. 
51. // These string constants contain the characters that the input: method accepts.
52. const NSString *Operators = @"+-*/";
53. const NSString *Equals    = @"=";
54. const NSString *Digits    = @"0123456789.";
55. const NSString *Period    = @".";
56. const NSString *Delete    = @"D";
57. const NSString *Clear     = @"C";
58. 
59. 
60. @implementation Calculator
61. 
62. #pragma mark Lifecycle
63. 
64. - init {
65.    if ((self = [super init])) {
66.       _display = [[NSMutableString stringWithCapacity:20] retain];
67.       _operator = nil;
68.    }
69.    return self;
70. }
71. - (void) dealloc {
72.    [_display release];
73.    [_operator release];
74.    [super dealloc];
75. }
76. 
77. 
78. #pragma mark Calculator Operation
79. 
80. /*
81.  * The input: method accepts the characters in the string constants
82.  * Operators, Equals, Digits, Period Delete, and Clear.
83.  *
84.  * The results of this method's computations are stored in _display.
85.  * This method uses _operand, and _operator in its calculations.
86.  */
87. - (void) input:(NSString *) input_character {
88.    static BOOL last_character_is_operator = NO;
89.    BOOL bad_character;
90. 
91.    // Does input_character contain exactly one character?
92.    if (!(bad_character = !(input_character && [input_character length] == 1))) {
93. 
94.       // Is input_character in Digits?
95.       if ([Digits rangeOfString: input_character].length) {
96.          if (last_character_is_operator) {
```

```
 97.             // Set the display to input_character.
 98.             [_display setString: input_character];
 99.
100.             last_character_is_operator = NO;
101.          }
102.          // Is input_character a digit, or is a period while a period has not been added to _display?
103.          else if (![input_character isEqualToString: (NSString *)Period] || [_display rangeOfString: (NSString *)Period].location ==
      NSNotFound) {
104.             // Add input_character to _display.
105.             [_display appendString:input_character];
106.          }
107.       }
108.
109.       // Is input_character in Operators or is it Equals?
110.       else if ([Operators rangeOfString:input_character].length || [input_character isEqualToString:(NSString *)Equals]) {
111.          if (!_operator && ![input_character isEqualToString:(NSString *)Equals]) {
112.             // input_character is this calculation's operator.
113.             //
114.             // Save the operand and the operator.
115.             _operand  = [[self displayValue] doubleValue];
116.             _operator = input_character;
117.          }
118.          else {
119.             // input_character is in Operators or Equals.
120.             //
121.             // Perform the computation indicated by the saved operator between the saved operand and _display.
122.             // Place the result in _display.
123.             if (_operator) {
124.                double operand2 = [[self displayValue] doubleValue];
125.                switch ([Operators rangeOfString: _operator].location) {
126.                   case 0:
127.                      _operand = _operand + operand2;
128.                      break;
129.                   case 1:
130.                      _operand = _operand - operand2;
131.                      break;
132.                   case 2:
133.                      _operand = _operand * operand2;
134.                      break;
135.                   case 3:
136.                      _operand = _operand / operand2;
137.                      break;
138.                }
139.                [_display setString: [[NSNumber numberWithDouble: _operand] stringValue]];
140.             }
141.             // Save the operation (if this is a chained computation).
142.             _operator = ([input_character isEqualToString:(NSString *)Equals])? nil : input_character;
143.          }
```

```
144.          last_character_is_operator = YES;
145.       }
146.       // Is input_character Delete?
147.       else if ([input_character isEqualToString:(NSString *)Delete]) {
148.          // Remove the rightmost character from _display.
149.          NSInteger index_of_char_to_remove = [_display length] - 1;
150.          if (index_of_char_to_remove >= 0) {
151.             [_display deleteCharactersInRange:NSMakeRange(index_of_char_to_remove, 1)];
152.             last_character_is_operator = NO;
153.          }
154.       }
155.       // Is input_character Clear?
156.       else if ([input_character isEqualToString:(NSString *)Clear]) {
157.          // If there's something in _display, clear it.
158.          if ([_display length]) {
159.             [_display setString:[NSString string]];
160.          }
161.          // Otherwise, clear the saved operator.
162.          else {
163.             _operator = nil;
164.          }
165.       }
166.       else {
167.          // input_character is an unexpected (invalid) character.
168.          bad_character = TRUE;
169.       }
170.    }
171.    if (bad_character) {
172.       // Raise exception for unexpected character.
173.       NSException *exception = [NSException exceptionWithName:NSInvalidArgumentException
174.                                                        reason:@"The input_character parameter contains an unexpected value."
175.                                                      userInfo:[NSDictionary dictionaryWithObjectsAndKeys: input_character, @"arg0", nil]];
176.       [exception raise];
177.    }
178. }
179.
180.
181. #pragma mark Outlets
182.
183. /*
184.  * The displayValue method rerutns a copy of _display.
185.  */
186. - (NSString *) displayValue {
187.    if ([_display length]) {
188.       return [[_display copy] autorelease];
189.    }
190.    return @"0";
191. }
```

```
192.
193. @end
```

```
/*
     File: CalculatorLogicTests.m
 Abstract: This file implements the logic-test suite for the Calculator class.
  Version: 2.0

 Disclaimer: IMPORTANT:  This Apple software is supplied to you by Apple
 Inc. ("Apple") in consideration of your agreement to the following
 terms, and your use, installation, modification or redistribution of
 this Apple software constitutes acceptance of these terms.  If you do
 not agree with these terms, please do not use, install, modify or
 redistribute this Apple software.

 In consideration of your agreement to abide by the following terms, and
 subject to these terms, Apple grants you a personal, non-exclusive
 license, under Apple's copyrights in this original Apple software (the
 "Apple Software"), to use, reproduce, modify and redistribute the Apple
 Software, with or without modifications, in source and/or binary forms;
 provided that if you redistribute the Apple Software in its entirety and
 without modifications, you must retain this notice and the following
 text and disclaimers in all such redistributions of the Apple Software.
 Neither the name, trademarks, service marks or logos of Apple Inc. may
 be used to endorse or promote products derived from the Apple Software
 without specific prior written permission from Apple.  Except as
 expressly stated in this notice, no other rights or licenses, express or
 implied, are granted by Apple herein, including but not limited to any
 patent rights that may be infringed by your derivative works or by other
 works in which the Apple Software may be incorporated.

 The Apple Software is provided by Apple on an "AS IS" basis.  APPLE
 MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION
 THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS
 FOR A PARTICULAR PURPOSE, REGARDING THE APPLE SOFTWARE OR ITS USE AND
 OPERATION ALONE OR IN COMBINATION WITH YOUR PRODUCTS.

 IN NO EVENT SHALL APPLE BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL
 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 INTERRUPTION) ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION,
 MODIFICATION AND/OR DISTRIBUTION OF THE APPLE SOFTWARE, HOWEVER CAUSED
 AND WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE),
 STRICT LIABILITY OR OTHERWISE, EVEN IF APPLE HAS BEEN ADVISED OF THE
 POSSIBILITY OF SUCH DAMAGE.

 Copyright (C) 2011 Apple Inc. All Rights Reserved.

 */

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

```
49.
50. #import "Calculator.h"
51.
52.
53. @interface CalculatorLogicTests : SenTestCase {
54. @private
55.     Calculator *calculator;
56. }
57.
58. @end
59.
60. @implementation CalculatorLogicTests
61.
62. /* The setUp method is called automatically before each test-case method (methods whose name starts with 'test').
63.  */
64. - (void) setUp {
65.     NSLog(@"%@ setUp", self.name);
66.     calculator = [[[Calculator alloc] init] retain];
67.     STAssertNotNil(calculator, @"Cannot create Calculator instance");
68. }
69.
70.
71. /* The tearDown method is called automatically after each test-case method (methods whose name starts with 'test').
72.  */
73. - (void) tearDown {
74.     [calculator release];
75.     NSLog(@"%@ tearDown", self.name);
76. }
77.
78. /* testAddition performs a simple addition test: 6 + 2 = 8.
79.  * The test has two parts:
80.  * 1. Through the input: method, feed the calculator the characters 6, +, 2, and =.
81.  * 2. Confirm that displayValue is 8.
82.  */
83. - (void) testAddition {
84.     NSLog(@"%@ start", self.name);   // self.name is the name of the test-case method.
85.     [calculator input:@"6"];
86.     [calculator input:@"+"];
87.     [calculator input:@"2"];
88.     [calculator input:@"="];
89.     STAssertTrue([[calculator displayValue] isEqualToString:@"8"], @"");
90.     NSLog(@"%@ end", self.name);
91. }
92.
93. /* testSubtraction performs a simple subtraction test: 19 - 2 = 17.
94.  * The test has two parts:
95.  * 1. Through the input: method, feed the calculator the characters 1, 9, -, 2, and =.
96.  * 2. Confirm that displayValue is 17.
```

```
 97.  */
 98. - (void) testSubtraction {
 99.    NSLog(@"%@ start", self.name);   // Use NSLog to generate additional build-results output.
100.    [calculator input:@"1"];
101.    [calculator input:@"9"];
102.    [calculator input:@"-"];
103.    [calculator input:@"2"];
104.    [calculator input:@"="];
105.    STAssertTrue([[calculator displayValue] isEqualToString:@"17"], @"");
106.    NSLog(@"%@ end", self.name);
107. }
108.
109. /* testDivision performs a simple division test: 19 / 8 = 2.375.
110.  * The test has two parts:
111.  * 1. Through the input: method, feed the calculator the characters 1, 9, /, 8, and =.
112.  * 2. Confirm that displayValue is 2.375.
113.  */
114. - (void) testDivision {
115.    NSLog(@"%@ start", self.name);
116.    [calculator input:@"1"];
117.    [calculator input:@"9"];
118.    [calculator input:@"/"];
119.    [calculator input:@"8"];
120.    [calculator input:@"="];
121.    STAssertTrue([[calculator displayValue] isEqualToString:@"2.375"], @"");
122.    NSLog(@"%@ end", self.name);
123. }
124.
125. /* testMultiplication performs a simple multiplication test: 6 * 2 = 12.
126.  * The test has two parts:
127.  * 1. Through the input: method, feed the calculator the characters 6, *, 2, and =.
128.  * 2. Confirm that displayValue is 12.
129.  */
130. - (void) testMultiplication {
131.    NSLog(@"%@ start", self.name);
132.    [calculator input:@"6"];
133.    [calculator input:@"*"];
134.    [calculator input:@"2"];
135.    [calculator input:@"="];
136.    STAssertTrue([[calculator displayValue] isEqualToString:@"12"], @"");
137.    NSLog(@"%@ end", self.name);
138. }
139.
140. /* testSubtractionNegativeResult performs a simple subtraction test with a negative result: 6 - 24 = -18.
141.  * The test has two parts:
142.  * 1. Through the input: method, feed the calculator the characters 6, -, 2, 4, and =.
143.  * 2. Confirm that displayValue is -18.
144.  */
```

```
145. - (void) testSubtractionNegativeResult {
146.    NSLog(@"%@ start", self.name);
147.    [calculator input:@"6"];
148.    [calculator input:@"-"];
149.    [calculator input:@"2"];
150.    [calculator input:@"4"];
151.    [calculator input:@"="];
152.    STAssertTrue([[calculator displayValue] isEqualToString:@"-18"], @"");
153.    NSLog(@"%@ end", self.name);
154. }
155.
156. /* testClearLastEntry ensures that the clear (C) key clears the last entry when used once.
157.  */
158. - (void) testClearLastEntry {
159.    NSLog(@"%@ start", self.name);
160.    [calculator input:@"7"];
161.    [calculator input:@"+"];
162.    [calculator input:@"3"];
163.    [calculator input:@"C"];
164.    [calculator input:@"4"];
165.    [calculator input:@"="];
166.    STAssertTrue([[calculator displayValue] isEqualToString:@"11"], @"");
167.    NSLog(@"%@ end", self.name);
168. }
169.
170. /* testClearComputation ensures that the clear (C) key clears the computation when used twice.
171.  */
172. - (void) testClearComputation {
173.    NSLog(@"%@ start", self.name);
174.    [calculator input:@"C"];
175.    [calculator input:@"7"];
176.    [calculator input:@"*"];
177.    [calculator input:@"3"];
178.    [calculator input:@"C"];
179.    [calculator input:@"C"];
180.    STAssertTrue([[calculator displayValue] isEqualToString:@"0"], @"");
181.    NSLog(@"%@ end", self.name);
182. }
183.
184. /* testInputException ensures that the input: method throws an exception in three situations:
185.  * 1. The argument contains more than one character.
186.  * 2. The argument contains an invalid character.
187.  * 3. The argument is nil.
188.  */
189. - (void) testInputException {
190.    NSLog(@"%@ start", self.name);
191.     STAssertThrows([calculator input:@"67"], @"No exception for multicharacter input.");
192.     STAssertThrows([calculator input:@"j"],  @"No exception for invalid input.");
```

```
193.      STAssertThrows([calculator input:nil],   @"No exception for nil input.");
194.     NSLog(@"%@ end", self.name);
195. }
196.
197. @end
```

```
/*
     File: iOS_CalcAppDelegate.h
 Abstract: This file declares the interface for the Calc application delegate.
  Version: 2.0

 Disclaimer: IMPORTANT:  This Apple software is supplied to you by Apple
 Inc. ("Apple") in consideration of your agreement to the following
 terms, and your use, installation, modification or redistribution of
 this Apple software constitutes acceptance of these terms.  If you do
 not agree with these terms, please do not use, install, modify or
 redistribute this Apple software.

 In consideration of your agreement to abide by the following terms, and
 subject to these terms, Apple grants you a personal, non-exclusive
 license, under Apple's copyrights in this original Apple software (the
 "Apple Software"), to use, reproduce, modify and redistribute the Apple
 Software, with or without modifications, in source and/or binary forms;
 provided that if you redistribute the Apple Software in its entirety and
 without modifications, you must retain this notice and the following
 text and disclaimers in all such redistributions of the Apple Software.
 Neither the name, trademarks, service marks or logos of Apple Inc. may
 be used to endorse or promote products derived from the Apple Software
 without specific prior written permission from Apple.  Except as
 expressly stated in this notice, no other rights or licenses, express or
 implied, are granted by Apple herein, including but not limited to any
 patent rights that may be infringed by your derivative works or by other
 works in which the Apple Software may be incorporated.

 The Apple Software is provided by Apple on an "AS IS" basis.  APPLE
 MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION
 THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS
 FOR A PARTICULAR PURPOSE, REGARDING THE APPLE SOFTWARE OR ITS USE AND
 OPERATION ALONE OR IN COMBINATION WITH YOUR PRODUCTS.

 IN NO EVENT SHALL APPLE BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL
 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 INTERRUPTION) ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION,
 MODIFICATION AND/OR DISTRIBUTION OF THE APPLE SOFTWARE, HOWEVER CAUSED
 AND WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE),
 STRICT LIABILITY OR OTHERWISE, EVEN IF APPLE HAS BEEN ADVISED OF THE
 POSSIBILITY OF SUCH DAMAGE.

 Copyright (C) 2011 Apple Inc. All Rights Reserved.

*/

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

```
49.
50. @class CalcViewController;
51.
52. @interface CalcAppDelegate : NSObject <UIApplicationDelegate> {
53.     UIWindow           *window;
54.     CalcViewController *calcViewController;
55. }
56.
57. @property (nonatomic, retain) IBOutlet UIWindow  *window;
58. @property (nonatomic, retain) CalcViewController *calcViewController;
59.
60. @end
```

```
/*
     File: iOS_CalcAppDelegate.m
 Abstract: This file implements the Calc application delegate.
  Version: 2.0

 Disclaimer: IMPORTANT:  This Apple software is supplied to you by Apple
 Inc. ("Apple") in consideration of your agreement to the following
 terms, and your use, installation, modification or redistribution of
 this Apple software constitutes acceptance of these terms.  If you do
 not agree with these terms, please do not use, install, modify or
 redistribute this Apple software.

 In consideration of your agreement to abide by the following terms, and
 subject to these terms, Apple grants you a personal, non-exclusive
 license, under Apple's copyrights in this original Apple software (the
 "Apple Software"), to use, reproduce, modify and redistribute the Apple
 Software, with or without modifications, in source and/or binary forms;
 provided that if you redistribute the Apple Software in its entirety and
 without modifications, you must retain this notice and the following
 text and disclaimers in all such redistributions of the Apple Software.
 Neither the name, trademarks, service marks or logos of Apple Inc. may
 be used to endorse or promote products derived from the Apple Software
 without specific prior written permission from Apple.  Except as
 expressly stated in this notice, no other rights or licenses, express or
 implied, are granted by Apple herein, including but not limited to any
 patent rights that may be infringed by your derivative works or by other
 works in which the Apple Software may be incorporated.

 The Apple Software is provided by Apple on an "AS IS" basis.  APPLE
 MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION
 THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS
 FOR A PARTICULAR PURPOSE, REGARDING THE APPLE SOFTWARE OR ITS USE AND
 OPERATION ALONE OR IN COMBINATION WITH YOUR PRODUCTS.

 IN NO EVENT SHALL APPLE BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL
 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 INTERRUPTION) ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION,
 MODIFICATION AND/OR DISTRIBUTION OF THE APPLE SOFTWARE, HOWEVER CAUSED
 AND WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE),
 STRICT LIABILITY OR OTHERWISE, EVEN IF APPLE HAS BEEN ADVISED OF THE
 POSSIBILITY OF SUCH DAMAGE.

 Copyright (C) 2011 Apple Inc. All Rights Reserved.

*/

#import "iOS_CalcAppDelegate.h"
```

```
49. #import "iOS_CalcViewController.h"
50. 
51. @implementation CalcAppDelegate
52. 
53. @synthesize window;
54. @synthesize calcViewController;
55. 
56. - (void) applicationDidFinishLaunching:(UIApplication *)application {
57. 
58.    // Override point for customization after application launch
59.    CalcViewController *_calcViewController = [[CalcViewController alloc] initWithNibName:@"CalcView" bundle:[NSBundle mainBundle]];
60.    self.calcViewController = _calcViewController;
61.    [window addSubview:[calcViewController view]];
62.    [window makeKeyAndVisible];
63. }
64. 
65. 
66. - (void) dealloc {
67.    [calcViewController release];
68.    [window release];
69.    [super dealloc];
70. }
71. 
72. - (void) applicationDidReceiveMemoryWarning:(UIApplication *)application {
73.    NSLog(@"Calc.CalcAppDelegate: applicationDidReceiveMemoryWarning:");
74. }
75. 
76. @end
```

```
/*
     File: iOS_CalcViewController.h
 Abstract: This file declares the interface for the CalcViewController class.
  Version: 2.0

 Disclaimer: IMPORTANT:  This Apple software is supplied to you by Apple
 Inc. ("Apple") in consideration of your agreement to the following
 terms, and your use, installation, modification or redistribution of
 this Apple software constitutes acceptance of these terms.  If you do
 not agree with these terms, please do not use, install, modify or
 redistribute this Apple software.

 In consideration of your agreement to abide by the following terms, and
 subject to these terms, Apple grants you a personal, non-exclusive
 license, under Apple's copyrights in this original Apple software (the
 "Apple Software"), to use, reproduce, modify and redistribute the Apple
 Software, with or without modifications, in source and/or binary forms;
 provided that if you redistribute the Apple Software in its entirety and
 without modifications, you must retain this notice and the following
 text and disclaimers in all such redistributions of the Apple Software.
 Neither the name, trademarks, service marks or logos of Apple Inc. may
 be used to endorse or promote products derived from the Apple Software
 without specific prior written permission from Apple.  Except as
 expressly stated in this notice, no other rights or licenses, express or
 implied, are granted by Apple herein, including but not limited to any
 patent rights that may be infringed by your derivative works or by other
 works in which the Apple Software may be incorporated.

 The Apple Software is provided by Apple on an "AS IS" basis.  APPLE
 MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION
 THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS
 FOR A PARTICULAR PURPOSE, REGARDING THE APPLE SOFTWARE OR ITS USE AND
 OPERATION ALONE OR IN COMBINATION WITH YOUR PRODUCTS.

 IN NO EVENT SHALL APPLE BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL
 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 INTERRUPTION) ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION,
 MODIFICATION AND/OR DISTRIBUTION OF THE APPLE SOFTWARE, HOWEVER CAUSED
 AND WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE),
 STRICT LIABILITY OR OTHERWISE, EVEN IF APPLE HAS BEEN ADVISED OF THE
 POSSIBILITY OF SUCH DAMAGE.

 Copyright (C) 2011 Apple Inc. All Rights Reserved.

*/

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

```
49. #import "Calculator.h"
50. 
51. @interface CalcViewController : UIViewController {
52.     id          displayField;
53.     Calculator *calculator;
54. }
55. 
56. @property (nonatomic, retain) IBOutlet id displayField;
57. 
58. - (IBAction) press:(id)sender;
59. 
60. @end
```

```
/*
     File: iOS_CalcViewController.m
 Abstract: This file implements the CalcViewController class.
  Version: 2.0

 Disclaimer: IMPORTANT:  This Apple software is supplied to you by Apple
 Inc. ("Apple") in consideration of your agreement to the following
 terms, and your use, installation, modification or redistribution of
 this Apple software constitutes acceptance of these terms.  If you do
 not agree with these terms, please do not use, install, modify or
 redistribute this Apple software.

 In consideration of your agreement to abide by the following terms, and
 subject to these terms, Apple grants you a personal, non-exclusive
 license, under Apple's copyrights in this original Apple software (the
 "Apple Software"), to use, reproduce, modify and redistribute the Apple
 Software, with or without modifications, in source and/or binary forms;
 provided that if you redistribute the Apple Software in its entirety and
 without modifications, you must retain this notice and the following
 text and disclaimers in all such redistributions of the Apple Software.
 Neither the name, trademarks, service marks or logos of Apple Inc. may
 be used to endorse or promote products derived from the Apple Software
 without specific prior written permission from Apple.  Except as
 expressly stated in this notice, no other rights or licenses, express or
 implied, are granted by Apple herein, including but not limited to any
 patent rights that may be infringed by your derivative works or by other
 works in which the Apple Software may be incorporated.

 The Apple Software is provided by Apple on an "AS IS" basis.  APPLE
 MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION
 THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS
 FOR A PARTICULAR PURPOSE, REGARDING THE APPLE SOFTWARE OR ITS USE AND
 OPERATION ALONE OR IN COMBINATION WITH YOUR PRODUCTS.

 IN NO EVENT SHALL APPLE BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL
 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 INTERRUPTION) ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION,
 MODIFICATION AND/OR DISTRIBUTION OF THE APPLE SOFTWARE, HOWEVER CAUSED
 AND WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE),
 STRICT LIABILITY OR OTHERWISE, EVEN IF APPLE HAS BEEN ADVISED OF THE
 POSSIBILITY OF SUCH DAMAGE.

 Copyright (C) 2011 Apple Inc. All Rights Reserved.

*/

#import "iOS_CalcViewController.h"
```

```
49.
50.
51. @implementation CalcViewController
52.
53. @synthesize displayField;
54.
55. - (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil {
56.    if ((self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil])) {
57.       calculator = [[Calculator alloc] init];
58.    }
59.    return self;
60. }
61.
62. - (BOOL) shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {
63.    return (interfaceOrientation == UIInterfaceOrientationPortrait);
64. }
65.
66. - (void) dealloc {
67.    [calculator release];
68.    [super dealloc];
69. }
70.
71. - (IBAction) press:(id)sender {
72.    [calculator input:[sender titleForState:UIControlStateNormal]];
73.    [displayField setText:[calculator displayValue]];
74. }
75.
76. @end
```

```
/*
     File: iOS_CalcApplicationTests.h
 Abstract: This file declares the interface for the logic-test suite for the Calculator class.
  Version: 2.0

 Disclaimer: IMPORTANT:  This Apple software is supplied to you by Apple
 Inc. ("Apple") in consideration of your agreement to the following
 terms, and your use, installation, modification or redistribution of
 this Apple software constitutes acceptance of these terms.  If you do
 not agree with these terms, please do not use, install, modify or
 redistribute this Apple software.

 In consideration of your agreement to abide by the following terms, and
 subject to these terms, Apple grants you a personal, non-exclusive
 license, under Apple's copyrights in this original Apple software (the
 "Apple Software"), to use, reproduce, modify and redistribute the Apple
 Software, with or without modifications, in source and/or binary forms;
 provided that if you redistribute the Apple Software in its entirety and
 without modifications, you must retain this notice and the following
 text and disclaimers in all such redistributions of the Apple Software.
 Neither the name, trademarks, service marks or logos of Apple Inc. may
 be used to endorse or promote products derived from the Apple Software
 without specific prior written permission from Apple.  Except as
 expressly stated in this notice, no other rights or licenses, express or
 implied, are granted by Apple herein, including but not limited to any
 patent rights that may be infringed by your derivative works or by other
 works in which the Apple Software may be incorporated.

 The Apple Software is provided by Apple on an "AS IS" basis.  APPLE
 MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION
 THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS
 FOR A PARTICULAR PURPOSE, REGARDING THE APPLE SOFTWARE OR ITS USE AND
 OPERATION ALONE OR IN COMBINATION WITH YOUR PRODUCTS.

 IN NO EVENT SHALL APPLE BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL
 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 INTERRUPTION) ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION,
 MODIFICATION AND/OR DISTRIBUTION OF THE APPLE SOFTWARE, HOWEVER CAUSED
 AND WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE),
 STRICT LIABILITY OR OTHERWISE, EVEN IF APPLE HAS BEEN ADVISED OF THE
 POSSIBILITY OF SUCH DAMAGE.

 Copyright (C) 2011 Apple Inc. All Rights Reserved.

 */

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

```
49.
50. #import <UIKit/UIKit.h>
51.
52. // Test-subject headers.
53. #import "iOS_CalcAppDelegate.h"
54. #import "iOS_CalcViewController.h"
55.
56.
57. @interface CalcApplicationTests : SenTestCase {
58. @private
59.     CalcAppDelegate    *app_delegate;
60.     CalcViewController *calc_view_controller;
61.     UIView             *calc_view;
62.
63. }
64.
65. @end
```

```
/*
     File: iOS_CalcApplicationTests.m
 Abstract: This file implements the logic-test suite for the Calculator class.
  Version: 2.0

 Disclaimer: IMPORTANT:  This Apple software is supplied to you by Apple
 Inc. ("Apple") in consideration of your agreement to the following
 terms, and your use, installation, modification or redistribution of
 this Apple software constitutes acceptance of these terms.  If you do
 not agree with these terms, please do not use, install, modify or
 redistribute this Apple software.

 In consideration of your agreement to abide by the following terms, and
 subject to these terms, Apple grants you a personal, non-exclusive
 license, under Apple's copyrights in this original Apple software (the
 "Apple Software"), to use, reproduce, modify and redistribute the Apple
 Software, with or without modifications, in source and/or binary forms;
 provided that if you redistribute the Apple Software in its entirety and
 without modifications, you must retain this notice and the following
 text and disclaimers in all such redistributions of the Apple Software.
 Neither the name, trademarks, service marks or logos of Apple Inc. may
 be used to endorse or promote products derived from the Apple Software
 without specific prior written permission from Apple.  Except as
 expressly stated in this notice, no other rights or licenses, express or
 implied, are granted by Apple herein, including but not limited to any
 patent rights that may be infringed by your derivative works or by other
 works in which the Apple Software may be incorporated.

 The Apple Software is provided by Apple on an "AS IS" basis.  APPLE
 MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION
 THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS
 FOR A PARTICULAR PURPOSE, REGARDING THE APPLE SOFTWARE OR ITS USE AND
 OPERATION ALONE OR IN COMBINATION WITH YOUR PRODUCTS.

 IN NO EVENT SHALL APPLE BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL
 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 INTERRUPTION) ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION,
 MODIFICATION AND/OR DISTRIBUTION OF THE APPLE SOFTWARE, HOWEVER CAUSED
 AND WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE),
 STRICT LIABILITY OR OTHERWISE, EVEN IF APPLE HAS BEEN ADVISED OF THE
 POSSIBILITY OF SUCH DAMAGE.

 Copyright (C) 2011 Apple Inc. All Rights Reserved.

 */


```

```
49. #import "iOS_CalcApplicationTests.h"
50.
51.
52. @implementation CalcApplicationTests
53.
54. /* The setUp method is called automatically for each test-case method (methods whose name starts with 'test').
55.  */
56. - (void) setUp {
57.     app_delegate         = [[UIApplication sharedApplication] delegate];
58.     calc_view_controller = app_delegate.calcViewController;
59.     calc_view            = calc_view_controller.view;
60. }
61.
62. - (void) testAppDelegate {
63.     STAssertNotNil(app_delegate, @"Cannot find the application delegate");
64. }
65.
66. /* testAddition performs a chained addition test.
67.  * The test has two parts:
68.  * 1. Check: 6 + 2 = 8.
69.  * 2. Check: display + 2 = 10.
70.  */
71. - (void) testAddition {
72.     [calc_view_controller press:[calc_view viewWithTag: 6]];  // 6
73.     [calc_view_controller press:[calc_view viewWithTag:13]];  // +
74.     [calc_view_controller press:[calc_view viewWithTag: 2]];  // 2
75.     [calc_view_controller press:[calc_view viewWithTag:12]];  // =
76.     STAssertTrue([[calc_view_controller.displayField text] isEqualToString:@"8"], @"Part 1 failed.");
77.
78.     [calc_view_controller press:[calc_view viewWithTag:13]];  // +
79.     [calc_view_controller press:[calc_view viewWithTag: 2]];  // 2
80.     [calc_view_controller press:[calc_view viewWithTag:12]];  // =
81.     STAssertTrue([[calc_view_controller.displayField text] isEqualToString:@"10"], @"Part 2 failed.");
82. }
83.
84. /* testSubtraction performs a simple subtraction test.
85.  * Check: 6 - 2 = 4.
86.  */
87. - (void) testSubtraction {
88.     [calc_view_controller press:[calc_view viewWithTag: 6]];  // 6
89.     [calc_view_controller press:[calc_view viewWithTag:14]];  // -
90.     [calc_view_controller press:[calc_view viewWithTag: 2]];  // 2
91.     [calc_view_controller press:[calc_view viewWithTag:12]];  // =
92.     STAssertTrue([[calc_view_controller.displayField text] isEqualToString:@"4"], @"");
93. }
94.
95. /* testDivision performs a simple division test.
96.  * Check: 25 / 4 = 6.25.
```

```
 97.   */
 98. - (void) testDivision {
 99.     [calc_view_controller press:[calc_view viewWithTag: 2]];  // 2
100.     [calc_view_controller press:[calc_view viewWithTag: 5]];  // 5
101.     [calc_view_controller press:[calc_view viewWithTag:16]];  // /
102.     [calc_view_controller press:[calc_view viewWithTag: 4]];  // 4
103.     [calc_view_controller press:[calc_view viewWithTag:12]];  // =
104.     STAssertTrue([[calc_view_controller.displayField text] isEqualToString:@"6.25"], @"");
105. }
106.
107. /* testMultiplication performs a simple multiplication test.
108.  * Check: 19 x 8 = 152.
109.  */
110. - (void) testMultiplication {
111.     [calc_view_controller press:[calc_view viewWithTag: 1]];  // 1
112.     [calc_view_controller press:[calc_view viewWithTag: 9]];  // 9
113.     [calc_view_controller press:[calc_view viewWithTag:15]];  // *
114.     [calc_view_controller press:[calc_view viewWithTag: 8]];  // 8
115.     [calc_view_controller press:[calc_view viewWithTag:12]];  // =
116.     STAssertTrue([[calc_view_controller.displayField text] isEqualToString:@"152"], @"");
117. }
118.
119. /* testDelete tests the functionality of the D (Delete) key.
120.  * 1. Enter the number 1987 into the calculator.
121.  * 2. Delete each digit, and test the display to ensure
122.  *    the correct display contains the expected value after each D press.
123.  */
124. - (void) testDelete {
125.     [calc_view_controller press:[calc_view viewWithTag: 1]];  // 1
126.     [calc_view_controller press:[calc_view viewWithTag: 9]];  // 9
127.     [calc_view_controller press:[calc_view viewWithTag: 8]];  // 8
128.     [calc_view_controller press:[calc_view viewWithTag: 7]];  // 7
129.     STAssertTrue([[calc_view_controller.displayField text] isEqualToString:@"1987"], @"Part 1 failed.");
130.
131.     [calc_view_controller press:[calc_view viewWithTag:19]];  // D (delete)
132.     STAssertTrue([[calc_view_controller.displayField text] isEqualToString:@"198"],  @"Part 2 failed.");
133.
134.     [calc_view_controller press:[calc_view viewWithTag:19]];  // D (delete)
135.     STAssertTrue([[calc_view_controller.displayField text] isEqualToString:@"19"],   @"Part 3 failed.");
136.
137.     [calc_view_controller press:[calc_view viewWithTag:19]];  // D (delete)
138.     STAssertTrue([[calc_view_controller.displayField text] isEqualToString:@"1"],    @"Part 4 failed.");
139.
140.     [calc_view_controller press:[calc_view viewWithTag:19]];  // D (delete)
141.     STAssertTrue([[calc_view_controller.displayField text] isEqualToString:@"0"],    @"Part 5 failed.");
142. }
143.
144. /* testClear tests the functionality of the C (Clear).
```

```
145.   * 1. Clear the display.
146.   *  - Enter the calculation 25 / 4.
147.   *  - Press C.
148.   *  - Ensure the display contains the value 0.
149.   * 2. Perform corrected computation.
150.   *  - Press 5, =.
151.   *  - Ensure the display contains the value 5.
152.   * 3. Ensure pressign C twice clears all.
153.   *  - Enter the calculation 19 x 8.
154.   *  - Press C (clears the display).
155.   *  - Press C (clears the operand).
156.   *  - Press +, 2, =.
157.   *  - Ensure the display contains the value 2.
158.   */
159. - (void) testClear {
160.     [calc_view_controller press:[calc_view viewWithTag: 2]];  // 2
161.     [calc_view_controller press:[calc_view viewWithTag: 5]];  // 5
162.     [calc_view_controller press:[calc_view viewWithTag:16]];  // /
163.     [calc_view_controller press:[calc_view viewWithTag: 4]];  // 4
164.     [calc_view_controller press:[calc_view viewWithTag:11]];  // C (clear)
165.     STAssertTrue([[calc_view_controller.displayField text] isEqualToString:@"0"], @"Part 1 failed.");
166.
167.     [calc_view_controller press:[calc_view viewWithTag: 5]];  // 5
168.     [calc_view_controller press:[calc_view viewWithTag:12]];  // =
169.     STAssertTrue([[calc_view_controller.displayField text] isEqualToString:@"5"], @"Part 2 failed.");
170.
171.     [calc_view_controller press:[calc_view viewWithTag: 1]];  // 1
172.     [calc_view_controller press:[calc_view viewWithTag: 9]];  // 9
173.     [calc_view_controller press:[calc_view viewWithTag:15]];  // x
174.     [calc_view_controller press:[calc_view viewWithTag: 8]];  // 8
175.     [calc_view_controller press:[calc_view viewWithTag:11]];  // C (clear)
176.     [calc_view_controller press:[calc_view viewWithTag:11]];  // C (all clear)
177.     [calc_view_controller press:[calc_view viewWithTag:13]];  // +
178.     [calc_view_controller press:[calc_view viewWithTag: 2]];  // 2
179.     [calc_view_controller press:[calc_view viewWithTag:12]];  // =
180.     STAssertTrue([[calc_view_controller.displayField text] isEqualToString:@"2"], @"Part 3 failed.");
181. }
182.
183. @end
```

iPhoneUnitTests illustrates the use of unit tests to ensure that an applicationÕs functionality does not degrade as its source code undergoes changes to improve the application or to fix bugs. The project showcases two types of unit tests: logic and application. Logic unit tests allow for stress-testing source code. Application unit tests help ensure the correct linkage between user-interface controls, controller objects, and model objects.

Minimum Buildtime Requirements:
- iOS SDK 4.3

Minimum Runtime Requirements:
- Simulator: iPhone/iPad 4.1 simulator
- Device: iOS 4.1

The iPhoneUnitTests project defines two schemes:
- iOS_Calc.   Runs the Calc application, and performs application unit tests on it.
- Calculator-iOS. Performs logic unit tests on the Calculator class.

The project contains four targets:
- iOS_Calc.   Builds the Calc application.
- iOS_Calc_ApplicationTests. Implements the application unit-test suite for the Calc application.
- Calculator-iOS.   Builds the Calculator-iOS static library.
- Calculator-iOS_LogicTests. Implements the logic unit-test suite for the Calculator class.

---
iOS_Calc Target
- This target builds an iPhone application (Calc) that implements a simple
arithmetic calculator.

iOS_Calc_ApplicationTests Target
- This target builds a unit-test bundle containing an application unit-test
suite (CalcApplicationTests) for the Calc application.

Calculator-iOS Target
- This target builds the static library that the Calc application uses to process its input and generate output to display to the user.
- The calculating engine is implemented in the Calculator class,
which has two main methods: input: and displayValue:
- The input: method accepts a one-character string as input, which represents a key press.
- The displayValue method provides the value representing the calculatorÕs output: As each key is pressed, the display value changes, as it would on a hardware-based calculator.

Calculator-iOS_LogicTests Target
- This target builds a unit-test bundle containing logic tests for the Calculator class.

Running Logic Tests on Calculator-iOS
- To run the logic tests:
  1. From the scheme toolbar menu, choose Calculator-iOS > iPhone 4.3 Simulator.
  2. Choose Product > Test. Xcode runs the test cases implemented in the CalculatorLogicTests.m file.
  3. Choose View > Navigators > Log to open the log navigator, containing the tests results.

```
  4. In the list on the left, select the Test Calculator-iOS_LogicTests session to view the test session log.

The results of the tests look similar to this:

GNU gdb 6.3.50-20050815 (Apple version gdb-1518) (Sat Feb 12 02:52:12 UTC 2011)
Copyright 2004 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
...
Test Suite 'CalculatorLogicTests' started at 2011-08-05 00:46:04 +0000
Test Case '-[CalculatorLogicTests testAddition]' started.
2011-08-04 17:46:04.333 otest[3858:903] -[CalculatorLogicTests testAddition] setUp
2011-08-04 17:46:04.334 otest[3858:903] -[CalculatorLogicTests testAddition] start
2011-08-04 17:46:04.337 otest[3858:903] -[CalculatorLogicTests testAddition] end
2011-08-04 17:46:04.338 otest[3858:903] -[CalculatorLogicTests testAddition] tearDown
Test Case '-[CalculatorLogicTests testAddition]' passed (0.005 seconds).
Test Case '-[CalculatorLogicTests testClearComputation]' started.
2011-08-04 17:46:04.338 otest[3858:903] -[CalculatorLogicTests testClearComputation] setUp
2011-08-04 17:46:04.339 otest[3858:903] -[CalculatorLogicTests testClearComputation] start
2011-08-04 17:46:04.340 otest[3858:903] -[CalculatorLogicTests testClearComputation] end
2011-08-04 17:46:04.340 otest[3858:903] -[CalculatorLogicTests testClearComputation] tearDown
Test Case '-[CalculatorLogicTests testClearComputation]' passed (0.002 seconds).
Test Case '-[CalculatorLogicTests testClearLastEntry]' started.
2011-08-04 17:46:04.341 otest[3858:903] -[CalculatorLogicTests testClearLastEntry] setUp
2011-08-04 17:46:04.341 otest[3858:903] -[CalculatorLogicTests testClearLastEntry] start
2011-08-04 17:46:04.342 otest[3858:903] -[CalculatorLogicTests testClearLastEntry] end
2011-08-04 17:46:04.342 otest[3858:903] -[CalculatorLogicTests testClearLastEntry] tearDown
Test Case '-[CalculatorLogicTests testClearLastEntry]' passed (0.002 seconds).
...
Test Suite 'CalculatorLogicTests' finished at 2011-08-05 00:46:04 +0000.
Executed 8 tests, with 0 failures (0 unexpected) in 0.033 (0.036) seconds
...
Executed 8 tests, with 0 failures (0 unexpected) in 0.033 (0.063) seconds

--------------------------------------------------------------------------------
Running Application Tests
- To run the application tests:
  1. From the scheme toolbar menu, choose iOS_Calc > <your_device>.
  2. Choose Product > Test. Xcode runs the test cases implemented in the iOS_CalcApplicationTests.m file.
  3. Choose View > Navigators > Log to open the log navigator.
  4. In the list on the left, select the Test iOS_Calc_ApplicationTests session to view the test session log.

- The results of the tests look similar to this:

GNU gdb 6.3.50-20050815 (Apple version gdb-1705) (Fri Jul  1 10:47:25 UTC 2011)
Copyright 2004 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
...
Test Suite 'All tests' started at 2011-08-05 02:02:50 +0000
```

```
 93. Test Suite '/Developer/Library/Frameworks/SenTestingKit.framework(Tests)' started at 2011-08-05 02:02:50 +0000
 94. Test Suite 'SenInterfaceTestCase' started at 2011-08-05 02:02:50 +0000
 95. Test Suite 'SenInterfaceTestCase' finished at 2011-08-05 02:02:50 +0000.
 96. Executed 0 tests, with 0 failures (0 unexpected) in 0.000 (0.002) seconds
 97.
 98. Test Suite '/Developer/Library/Frameworks/SenTestingKit.framework(Tests)' finished at 2011-08-05 02:02:50 +0000.
 99. Executed 0 tests, with 0 failures (0 unexpected) in 0.000 (0.009) seconds
100.
101. Test Suite '/var/mobile/Applications/C3898898-E45A-437E-B41C-122A91075031/iOS_Calc_ApplicationTests.octest(Tests)' started at 2011-08-05
     02:02:50 +0000
102. Test Suite 'CalcApplicationTests' started at 2011-08-05 02:02:50 +0000
103. Test Case '-[CalcApplicationTests testAddition]' started.
104. Test Case '-[CalcApplicationTests testAddition]' passed (0.007 seconds).
105. Test Case '-[CalcApplicationTests testAppDelegate]' started.
106. Test Case '-[CalcApplicationTests testAppDelegate]' passed (0.001 seconds).
107. Test Case '-[CalcApplicationTests testClear]' started.
108. Test Case '-[CalcApplicationTests testClear]' passed (0.004 seconds).
109. Test Case '-[CalcApplicationTests testDelete]' started.
110. Test Case '-[CalcApplicationTests testDelete]' passed (0.002 seconds).
111. Test Case '-[CalcApplicationTests testDivision]' started.
112. Test Case '-[CalcApplicationTests testDivision]' passed (0.003 seconds).
113. Test Case '-[CalcApplicationTests testMultiplication]' started.
114. Test Case '-[CalcApplicationTests testMultiplication]' passed (0.002 seconds).
115. Test Case '-[CalcApplicationTests testSubtraction]' started.
116. Test Case '-[CalcApplicationTests testSubtraction]' passed (0.002 seconds).
117. Test Suite 'CalcApplicationTests' finished at 2011-08-05 02:02:50 +0000.
118. Executed 7 tests, with 0 failures (0 unexpected) in 0.020 (0.031) seconds
119.
120. Test Suite '/var/mobile/Applications/C3898898-E45A-437E-B41C-122A91075031/iOS_Calc_ApplicationTests.octest(Tests)' finished at 2011-08-05
     02:02:50 +0000.
121. Executed 7 tests, with 0 failures (0 unexpected) in 0.020 (0.037) seconds
122.
123. Test Suite 'All tests' finished at 2011-08-05 02:02:50 +0000.
124. Executed 7 tests, with 0 failures (0 unexpected) in 0.020 (0.061) seconds
125.
126. --------------------------------------------------------------------------------
127. Related Information
128. - For more information, see the ÒUnit Testing ApplicationsÓ chapter in the iOS Development Workflow Guide.
129.
130. Version 2.0
131. - Updated for Xcode 4.0.2 and iOS SDK 4.3.
132.
133. Version 1.2
134. - Fixed bugs. Added workaround for running unit tests against the iPhone Simulator in Xcode 3.2.4 with iOS SDK 4.1.
135.
136. Version 1.1
137. - Upgraded project to build with the iOS 4 SDK.
138.
```

```
Version 1.0
 - First Version

Copyright © 2011 Apple Inc. All rights reserved.
```

```
1.  //
2.  //  AppDelegate.h
3.  //  MasterDetail
4.  //
5.  //  David J. Malan
6.  //  Harvard University
7.  //  malan@harvard.edu
8.  //
9.  //  Demonstrates a Master-Detail Application.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @interface AppDelegate : UIResponder <UIApplicationDelegate>
15.
16. @property (strong, nonatomic) UINavigationController *navigationController;
17. @property (strong, nonatomic) UIWindow *window;
18.
19. @end
```

```
//
//  AppDelegate.m
//  MasterDetail
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates a Master-Detail Application.
//

#import "AppDelegate.h"

#import "MasterViewController.h"

@implementation AppDelegate

@synthesize navigationController = _navigationController;
@synthesize window = _window;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    MasterViewController *masterViewController = [[MasterViewController alloc] initWithNibName:@"MasterViewController" bundle:nil];
    self.navigationController = [[UINavigationController alloc] initWithRootViewController:masterViewController];
    self.window.rootViewController = self.navigationController;
    [self.window makeKeyAndVisible];
    return YES;
}

@end
```

```
1. //
2. //  DetailViewController.h
3. //  MasterDetail
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  malan@harvard.edu
8. //
9. //  Demonstrates a Master-Detail Application.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @interface DetailViewController : UIViewController
15.
16. @property (strong, nonatomic) IBOutlet UILabel *detailDescriptionLabel;
17. @property (strong, nonatomic) id detailItem;
18.
19. @end
```

```
//
//  DetailViewController.m
//  MasterDetail
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates a Master-Detail Application.
//

#import "DetailViewController.h"

@interface DetailViewController ()
- (void)configureView;
@end

@implementation DetailViewController

@synthesize detailItem = _detailItem;
@synthesize detailDescriptionLabel = _detailDescriptionLabel;

#pragma mark - Managing the detail item

- (void)setDetailItem:(id)newDetailItem
{
    if (_detailItem != newDetailItem) {
        _detailItem = newDetailItem;

        // Update the view.
        [self configureView];
    }
}

- (void)configureView
{
    // Update the user interface for the detail item.
    if (self.detailItem) {
        self.detailDescriptionLabel.text = [self.detailItem description];
    }
}

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

```

```
49. - (void)viewDidUnload
50. {
51.     [super viewDidUnload];
52.     self.detailDescriptionLabel = nil;
53. }
54. 
55. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
56. {
57.     return (interfaceOrientation != UIInterfaceOrientationPortraitUpsideDown);
58. }
59. 
60. - (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
61. {
62.     self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
63.     if (self) {
64.         self.title = NSLocalizedString(@"Detail", @"Detail");
65.     }
66.     return self;
67. }
68. 
69. @end
```

```
1.  //
2.  //  MasterViewController.h
3.  //  MasterDetail
4.  //
5.  //  David J. Malan
6.  //  Harvard University
7.  //  malan@harvard.edu
8.  //
9.  //  Demonstrates a Master-Detail Application.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @class DetailViewController;
15.
16. @interface MasterViewController : UITableViewController
17.
18. @property (strong, nonatomic) DetailViewController *detailViewController;
19.
20. @end
```

```
1. //
2. //  MasterViewController.m
3. //  MasterDetail
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  malan@harvard.edu
8. //
9. //  Demonstrates a Master-Detail Application.
10. //
11.
12. #import "MasterViewController.h"
13.
14. #import "DetailViewController.h"
15.
16. @interface MasterViewController () {
17.     NSMutableArray *_objects;
18. }
19. @end
20.
21. @implementation MasterViewController
22.
23. @synthesize detailViewController = _detailViewController;
24.
25. - (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
26. {
27.     self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
28.     if (self) {
29.         self.title = NSLocalizedString(@"Master", @"Master");
30.     }
31.     return self;
32. }
33.
34. - (void)viewDidLoad
35. {
36.     [super viewDidLoad];
37.     self.navigationItem.leftBarButtonItem = self.editButtonItem;
38.     UIBarButtonItem *addButton = [[UIBarButtonItem alloc] initWithBarButtonSystemItem:UIBarButtonSystemItemAdd target:self action:@selector(
    insertNewObject:)];
39.     self.navigationItem.rightBarButtonItem = addButton;
40. }
41.
42. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
43. {
44.     return (interfaceOrientation != UIInterfaceOrientationPortraitUpsideDown);
45. }
46.
47. - (void)insertNewObject:(id)sender
```

```
48. {
49.     if (!_objects) {
50.         _objects = [[NSMutableArray alloc] init];
51.     }
52.     [_objects insertObject:[NSDate date] atIndex:0];
53.     NSIndexPath *indexPath = [NSIndexPath indexPathForRow:0 inSection:0];
54.     [self.tableView insertRowsAtIndexPaths:[NSArray arrayWithObject:indexPath] withRowAnimation:UITableViewRowAnimationAutomatic];
55. }
56.
57. #pragma mark - Table View
58.
59. - (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
60. {
61.     return 1;
62. }
63.
64. - (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
65. {
66.     return _objects.count;
67. }
68.
69. // Customize the appearance of table view cells.
70. - (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
71. {
72.     static NSString *CellIdentifier = @"Cell";
73.
74.     UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];
75.     if (cell == nil) {
76.         cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:CellIdentifier];
77.         cell.accessoryType = UITableViewCellAccessoryDisclosureIndicator;
78.     }
79.
80.     NSDate *object = [_objects objectAtIndex:indexPath.row];
81.     cell.textLabel.text = [object description];
82.     return cell;
83. }
84.
85. - (BOOL)tableView:(UITableView *)tableView canEditRowAtIndexPath:(NSIndexPath *)indexPath
86. {
87.     // Return NO if you do not want the specified item to be editable.
88.     return YES;
89. }
90.
91. - (void)tableView:(UITableView *)tableView commitEditingStyle:(UITableViewCellEditingStyle)editingStyle forRowAtIndexPath:(NSIndexPath *)
    indexPath
92. {
93.     if (editingStyle == UITableViewCellEditingStyleDelete) {
94.         [_objects removeObjectAtIndex:indexPath.row];
```

```
 95.         [tableView deleteRowsAtIndexPaths:[NSArray arrayWithObject:indexPath] withRowAnimation:UITableViewRowAnimationFade];
 96.     }
 97. }
 98.
 99. - (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath
100. {
101.     if (!self.detailViewController) {
102.         self.detailViewController = [[DetailViewController alloc] initWithNibName:@"DetailViewController" bundle:nil];
103.     }
104.     NSDate *object = [_objects objectAtIndex:indexPath.row];
105.     self.detailViewController.detailItem = object;
106.     [self.navigationController pushViewController:self.detailViewController animated:YES];
107. }
108.
109. @end
```

```
//
//  AppDelegate.h
//  Paddle
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates Core Graphics with a rectangle.
//

#import <UIKit/UIKit.h>

@class ViewController;

@interface AppDelegate : UIResponder <UIApplicationDelegate>

@property (strong, nonatomic) ViewController *viewController;
@property (strong, nonatomic) UIWindow *window;

@end
```

```m
//
//  AppDelegate.m
//  Paddle
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates Core Graphics with a rectangle.
//

#import "AppDelegate.h"
#import "ViewController.h"

@implementation AppDelegate

@synthesize viewController=_viewController;
@synthesize window=_window;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    self.viewController = [[ViewController alloc] initWithNibName:@"ViewController" bundle:nil];
    self.window.rootViewController = self.viewController;
    [self.window makeKeyAndVisible];
    return YES;
}

@end
```

```
1.  //
2.  //  PaddleView.h
3.  //  Paddle
4.  //
5.  //  David J. Malan
6.  //  Harvard University
7.  //  malan@harvard.edu
8.  //
9.  //  Demonstrates Core Graphics with a rectangle.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @interface PaddleView : UIView
15. @end
```

```m
//
//  PaddleView.m
//  Paddle
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates Core Graphics with a rectangle.
//

#import "PaddleView.h"

@implementation PaddleView

- (void)drawRect:(CGRect)rect
{
    CGRect square = CGRectMake(0.0f, 0.0f, 10.0f, 60.0f);
    [[UIColor whiteColor] set];
    UIRectFill(square);
}

@end
```

```
1.  //
2.  //  ViewController.h
3.  //  Paddle
4.  //
5.  //  David J. Malan
6.  //  Harvard University
7.  //  malan@harvard.edu
8.  //
9.  //  Demonstrates Core Graphics with a rectangle.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @interface ViewController : UIViewController
15. @end
```

```
1.  //
2.  //  ViewController.m
3.  //  Paddle
4.  //
5.  //  David J. Malan
6.  //  Harvard University
7.  //  malan@harvard.edu
8.  //
9.  //  Demonstrates Core Graphics with a rectangle.
10. //
11.
12. #import "ViewController.h"
13. #import "PaddleView.h"
14.
15.
16. @interface ViewController ()
17.
18. // private property
19. @property (nonatomic, retain) PaddleView *paddleView;
20.
21. @end
22.
23.
24. @implementation ViewController
25.
26. @synthesize paddleView=_paddleView;
27.
28. - (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
29. {
30.     if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil])
31.     {
32.         // add paddle
33.         self.paddleView = [[PaddleView alloc] initWithFrame:CGRectMake(10.0f, 10.0f, 10.0f, 60.0f)];
34.         [self.view addSubview:self.paddleView];
35.     }
36.     return self;
37. }
38.
39. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
40. {
41.     return (interfaceOrientation == UIInterfaceOrientationLandscapeRight);
42. }
43.
44. - (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event
45. {
46.     // follow user's finger vertically with paddle
47.     UITouch *touch = [[event allTouches] anyObject];
48.     CGPoint location = [touch locationInView:self.view];
```

```
49.     self.paddleView.center = CGPointMake(self.paddleView.center.x, location.y);
50. }
51.
52. @end
```

```
1. //
2. //  AppDelegate.h
3. //  Plist
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  malan@harvard.edu
8. //
9. //  Demonstrates property lists.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @class ViewController;
15.
16. @interface AppDelegate : UIResponder <UIApplicationDelegate>
17.
18. @property (strong, nonatomic) ViewController *viewController;
19. @property (strong, nonatomic) UIWindow *window;
20.
21. @end
```

```
1.  //
2.  //  AppDelegate.m
3.  //  Plist
4.  //
5.  //  David J. Malan
6.  //  Harvard University
7.  //  malan@harvard.edu
8.  //
9.  //  Demonstrates property lists.
10. //
11.
12. #import "AppDelegate.h"
13. #import "ViewController.h"
14.
15. @implementation AppDelegate
16.
17. @synthesize viewController=_viewController;
18. @synthesize window=_window;
19.
20. - (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
21. {
22.     self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
23.     self.viewController = [[ViewController alloc] initWithNibName:@"ViewController" bundle:nil];
24.     self.window.rootViewController = self.viewController;
25.     [self.window makeKeyAndVisible];
26.     return YES;
27. }
28.
29. @end
```

```
1. //
2. //  ViewController.h
3. //  Plist
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  malan@harvard.edu
8. //
9. //  Demonstrates property lists.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @interface ViewController : UIViewController
15. @end
```

```m
//
//  ViewController.m
//  Plist
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates property lists.
//

#import "ViewController.h"


@interface ViewController ()

// private property
@property (nonatomic, readwrite, strong) NSArray *words;

@end


@implementation ViewController

@synthesize words=_words;

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil])
    {
        // load words
        NSString *path = [[NSBundle mainBundle] pathForResource:@"small" ofType:@"plist"];
        NSArray *words = [[NSArray alloc] initWithContentsOfFile:path];
        self.words = words;
    }
    return self;
}

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return 1;
}

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
{
    return (interfaceOrientation == UIInterfaceOrientationPortrait);
}

```

```
49. - (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
50. {
51.     // allocate cell, reusing if possible
52.     static NSString *CellIdentifier = @"Cell";
53.     UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];
54.     if (cell == nil)
55.         cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:CellIdentifier];
56.
57.     // configure cell
58.     cell.selectionStyle = UITableViewCellSelectionStyleNone;
59.     cell.textLabel.text = [self.words objectAtIndex:[indexPath row]];
60.
61.     return cell;
62. }
63.
64. - (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
65. {
66.     return [self.words count];
67. }
68.
69. @end
```

```
//
//  AppDelegate.h
//  Pong
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates Core Graphics.
//  Inspired by iTennis by Brandon Trebitowski.
//

#import <UIKit/UIKit.h>

@class ViewController;

@interface AppDelegate : UIResponder <UIApplicationDelegate>

@property (strong, nonatomic) ViewController *viewController;
@property (strong, nonatomic) UIWindow *window;

@end
```

```m
//
//  AppDelegate.m
//  Pong
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates Core Graphics.
//  Inspired by iTennis by Brandon Trebitowski.
//

#import "AppDelegate.h"

#import "ViewController.h"

@implementation AppDelegate

@synthesize viewController=_viewController;
@synthesize window=_window;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    self.viewController = [[ViewController alloc] initWithNibName:@"ViewController" bundle:nil];
    self.window.rootViewController = self.viewController;
    [self.window makeKeyAndVisible];
    return YES;
}

- (void)applicationWillResignActive:(UIApplication *)application
{
    [self.viewController kickoff];
}

@end
```

```
//
//  PongView.h
//  Pong
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates Core Graphics.
//  Inspired by iTennis by Brandon Trebitowski.
//

#import <UIKit/UIKit.h>

@interface PongView : UIView
@end
```

```
//
//  PongView.m
//  Pong
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates Core Graphics.
//  Inspired by iTennis by Brandon Trebitowski.
//

#import "PongView.h"

@implementation PongView

- (void)drawRect:(CGRect)rect
{
    // get context
    CGContextRef context = UIGraphicsGetCurrentContext();

    // draw dashed midfield line
    CGFloat dashes[] = {1,1};
    CGContextSetLineDash(context, 0.0, dashes, 2);
    CGContextSetStrokeColorWithColor(context, [UIColor whiteColor].CGColor);
    CGContextSetLineWidth(context, 5.0f);
    CGContextMoveToPoint(context, 240.0f, 0.0f);
    CGContextAddLineToPoint(context, 240.0f, 320.0f);
    CGContextStrokePath(context);
}

@end
```

```
//
//  ViewController.h
//  Pong
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates Core Graphics.
//  Inspired by iTennis by Brandon Trebitowski.
//

#import <UIKit/UIKit.h>

@interface ViewController : UIViewController
- (void)kickoff;
@end
```

```
1. //
2. //  ViewController.m
3. //  Pong
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  malan@harvard.edu
8. //
9. //  Demonstrates Core Graphics.
10. //  Inspired by iTennis by Brandon Trebitowski.
11. //
12.
13. #import "ViewController.h"
14.
15.
16. // default velocity
17. static const float VELOCITY = 10.0f;
18.
19.
20. @interface ViewController () {
21. @private
22. CGPoint _velocity;
23. }
24.
25. // private properties
26. @property (nonatomic, readwrite, weak) IBOutlet UIImageView *ball;
27. @property (nonatomic, readwrite, weak) IBOutlet UILabel *labelLeft;
28. @property (nonatomic, readwrite, weak) IBOutlet UILabel *labelRight;
29. @property (nonatomic, readwrite, weak) IBOutlet UIImageView *paddleLeft;
30. @property (nonatomic, readwrite, weak) IBOutlet UIImageView *paddleRight;
31. @property (assign, nonatomic, readwrite) BOOL paused;
32. @property (assign, nonatomic, readwrite) NSUInteger scoreLeft;
33. @property (assign, nonatomic, readwrite) NSUInteger scoreRight;
34.
35. @end
36.
37.
38. @implementation ViewController
39.
40. @synthesize ball=_ball;
41. @synthesize labelLeft=_labelLeft;
42. @synthesize labelRight=_labelRight;
43. @synthesize paddleLeft=_paddleLeft;
44. @synthesize paddleRight=_paddleRight;
45. @synthesize paused=_paused;
46. @synthesize scoreLeft=_scoreLeft;
47. @synthesize scoreRight=_scoreRight;
48.
```

```
49.
50. - (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
51. {
52.     if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil])
53.     {
54.         // initialize scores
55.         self.scoreLeft = 0;
56.         self.scoreRight = 0;
57.
58.         // initialize ball's velocity
59.         _velocity = CGPointMake(VELOCITY, VELOCITY);
60.
61.         // schedule movement
62.         [NSTimer scheduledTimerWithTimeInterval:0.05 target:self selector:@selector(play) userInfo:nil repeats:YES];
63.     }
64.     return self;
65. }
66.
67. - (void)kickoff
68. {
69.     // pause
70.     self.paused = YES;
71.
72.     // update scores
73.     self.labelLeft.text = [NSString stringWithFormat:@"%u", self.scoreLeft];
74.     self.labelRight.text = [NSString stringWithFormat:@"%u", self.scoreRight];
75.
76.     // center ball
77.     self.ball.center = CGPointMake(240.0f, 160.0f);
78.
79.     // align paddles
80.     self.paddleLeft.center = CGPointMake(25.0f, 160.0f);
81.     self.paddleRight.center = CGPointMake(455.0f, 160.0f);
82. }
83.
84. - (void)play
85. {
86.     // check whether paused
87.     if (self.paused)
88.         return;
89.
90.     // move ball
91.     self.ball.center = CGPointMake(self.ball.center.x + _velocity.x, self.ball.center.y + _velocity.y);
92.
93.     // detect goals
94.     if (self.ball.center.x < 5) {
95.         self.scoreRight++;
96.         [self kickoff];
```

```
 97.     }
 98.     else if (self.view.bounds.size.width - 5 < self.ball.center.x) {
 99.         self.scoreLeft++;
100.         [self kickoff];
101.     }
102. 
103.     // bounce off of top and bottom walls
104.     if (self.ball.center.y < 5 || self.view.bounds.size.height - 5 < self.ball.center.y) {
105.         _velocity.y = -_velocity.y;
106.     }
107. 
108.     // bounce off of left paddle
109.     if (CGRectIntersectsRect(self.ball.frame, self.paddleLeft.frame)) {
110.         if (self.paddleLeft.center.x < self.ball.center.x) {
111.             _velocity.x = -_velocity.x;
112.         }
113.     }
114. 
115.     // bounce off of right paddle
116.     if (CGRectIntersectsRect(self.ball.frame, self.paddleRight.frame)) {
117.         if (self.ball.center.x < self.paddleRight.center.x) {
118.             _velocity.x = -_velocity.x;
119.         }
120.     }
121. 
122.     // move opponent as ball approaches
123.     if (_velocity.x < 0) {
124. 
125.         // move down
126.         if (self.ball.center.y < self.paddleLeft.center.y) {
127.             self.paddleLeft.center = CGPointMake(self.paddleLeft.center.x, self.paddleLeft.center.y - 10.0f);
128.         }
129. 
130.         // move up
131.         else if (self.ball.center.y > self.paddleLeft.center.y) {
132.             self.paddleLeft.center = CGPointMake(self.paddleLeft.center.x, self.paddleLeft.center.y + 10.0f);
133.         }
134.     }
135. }
136. 
137. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
138. {
139.     return (interfaceOrientation == UIInterfaceOrientationLandscapeRight);
140. }
141. 
142. - (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
143. {
144.     // unpause if paused
```

```
145.     if (self.paused)
146.         self.paused = NO;
147. }
148.
149. - (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event
150. {
151.     // follow user's finger vertically with left paddle
152.     UITouch *touch = [[event allTouches] anyObject];
153.     CGPoint location = [touch locationInView:self.view];
154.     self.paddleRight.center = CGPointMake(self.paddleRight.center.x, location.y);
155. }
156.
157. - (void)viewDidAppear:(BOOL)animated
158. {
159.     // play ball!
160.     [self kickoff];
161. }
162.
163. @end
```

```
1.  //
2.  //  AppDelegate.h
3.  //  Sqlite
4.  //
5.  //  David J. Malan
6.  //  Harvard University
7.  //  malan@harvard.edu
8.  //
9.  //  Demonstrates SQLite.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @class ViewController;
15.
16. @interface AppDelegate : UIResponder <UIApplicationDelegate>
17.
18. @property (strong, nonatomic) ViewController *viewController;
19. @property (strong, nonatomic) UIWindow *window;
20.
21. @end
```

```m
//
//  AppDelegate.m
//  Sqlite
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates SQLite.
//

#import "AppDelegate.h"

#import "ViewController.h"

@implementation AppDelegate

@synthesize window=_window;
@synthesize viewController=_viewController;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    self.viewController = [[ViewController alloc] initWithNibName:@"ViewController" bundle:nil];
    self.window.rootViewController = self.viewController;
    [self.window makeKeyAndVisible];
    return YES;
}

@end
```

```
1. //
2. //  ViewController.h
3. //  Sqlite
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  malan@harvard.edu
8. //
9. //  Demonstrates SQLite.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @interface ViewController : UIViewController
15. @end
```

```m
//
//  ViewController.m
//  Sqlite
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates SQLite.
//

#import "ViewController.h"
#import "sqlite3.h"


@interface ViewController ()

// private property
@property (nonatomic, readwrite, strong) NSMutableArray *words;

@end


@implementation ViewController

@synthesize words=_words;

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil])
    {
        // prepare for words
        self.words = [[NSMutableArray alloc] init];

        // connect to SQLite database
        sqlite3 *db;
        NSString *path = [[NSBundle mainBundle] pathForResource:@"small" ofType:@"sqlite"];
        sqlite3_open([path UTF8String], &db);

        // select all words
        NSString *sql = @"SELECT word FROM words";
        sqlite3_stmt *statement;
        sqlite3_prepare_v2(db, [sql UTF8String], -1, &statement, nil);

        // iterate over results
        while (sqlite3_step(statement) == SQLITE_ROW) {
            char *c = (char *) sqlite3_column_text(statement, 0);
            NSString *s = [[NSString alloc] initWithUTF8String:c];
```

```
            [self.words addObject:s];
        }

        // close database
        sqlite3_close(db);
    }
    return self;
}

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return 1;
}

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
{
    return (interfaceOrientation == UIInterfaceOrientationPortrait);
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    // allocate cell, reusing if possible
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];
    if (cell == nil)
        cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:CellIdentifier];

    // configure cell
    cell.selectionStyle = UITableViewCellSelectionStyleNone;
    cell.textLabel.text = [self.words objectAtIndex:[indexPath row]];

    return cell;
}

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

@end
```

```
1.  //
2.  //  AppDelegate.h
3.  //  Tabbed
4.  //
5.  //  David J. Malan
6.  //  Harvard University
7.  //  malan@harvard.edu
8.  //
9.  //  Demonstrates a Tabbed Application.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @interface AppDelegate : UIResponder <UIApplicationDelegate, UITabBarControllerDelegate>
15.
16. @property (strong, nonatomic) UITabBarController *tabBarController;
17. @property (strong, nonatomic) UIWindow *window;
18.
19. @end
```

```
1.  //
2.  //  AppDelegate.h
3.  //  Tabbed
4.  //
5.  //  David J. Malan
6.  //  Harvard University
7.  //  malan@harvard.edu
8.  //
9.  //  Demonstrates a Tabbed Application.
10. //
11.
12. #import "AppDelegate.h"
13. #import "FirstViewController.h"
14. #import "SecondViewController.h"
15.
16. @implementation AppDelegate
17.
18. @synthesize tabBarController = _tabBarController;
19. @synthesize window = _window;
20.
21. - (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
22. {
23.     self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
24.     UIViewController *viewController1 = [[FirstViewController alloc] initWithNibName:@"FirstViewController" bundle:nil];
25.     UIViewController *viewController2 = [[SecondViewController alloc] initWithNibName:@"SecondViewController" bundle:nil];
26.     self.tabBarController = [[UITabBarController alloc] init];
27.     self.tabBarController.viewControllers = [NSArray arrayWithObjects:viewController1, viewController2, nil];
28.     self.window.rootViewController = self.tabBarController;
29.     [self.window makeKeyAndVisible];
30.     return YES;
31. }
32.
33. @end
```

```
1. //
2. //  FirstViewController.h
3. //  Tabbed
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  malan@harvard.edu
8. //
9. //  Demonstrates a Tabbed Application.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @interface FirstViewController : UIViewController
15. @end
```

```m
//
//  FirstViewController.m
//  Tabbed
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates a Tabbed Application.
//

#import "FirstViewController.h"

@implementation FirstViewController

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    if (self) {
        self.title = NSLocalizedString(@"First", @"First");
        self.tabBarItem.image = [UIImage imageNamed:@"first"];
    }
    return self;
}

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
{
    return (interfaceOrientation != UIInterfaceOrientationPortraitUpsideDown);
}

@end
```

```
1. //
2. //  SecondViewController.h
3. //  Tabbed
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  malan@harvard.edu
8. //
9. //  Demonstrates a Tabbed Application.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @interface SecondViewController : UIViewController
15. @end
```

```m
//
//  SecondViewController.m
//  Tabbed
//
//  David J. Malan
//  Harvard University
//  malan@harvard.edu
//
//  Demonstrates a Tabbed Application.
//

#import "SecondViewController.h"

@implementation SecondViewController

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    if (self) {
        self.title = NSLocalizedString(@"Second", @"Second");
        self.tabBarItem.image = [UIImage imageNamed:@"second"];
    }
    return self;
}

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
{
    return (interfaceOrientation != UIInterfaceOrientationPortraitUpsideDown);
}

@end
```