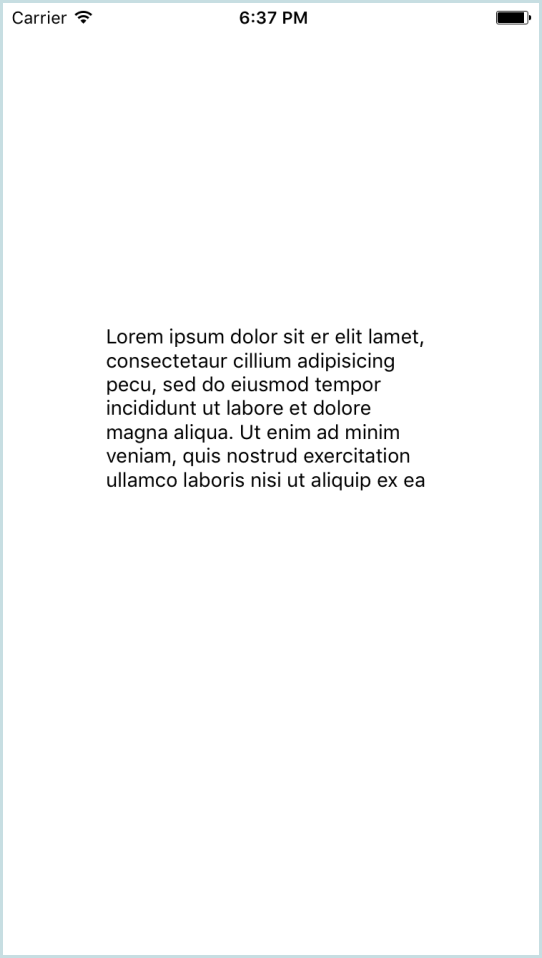
TDD workshop

Agenda

- Hello World
- Benefits of TDD
- Code coverage
- Roman Numerals Kata
- Zodiac app
- ETA Detroit app

Hello World


Carrier
6:37 PM
Lorem ipsum dolor sit er elit lamet, consectetaur cillium adipisicing pecu, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea

Lab 1 - Hello World

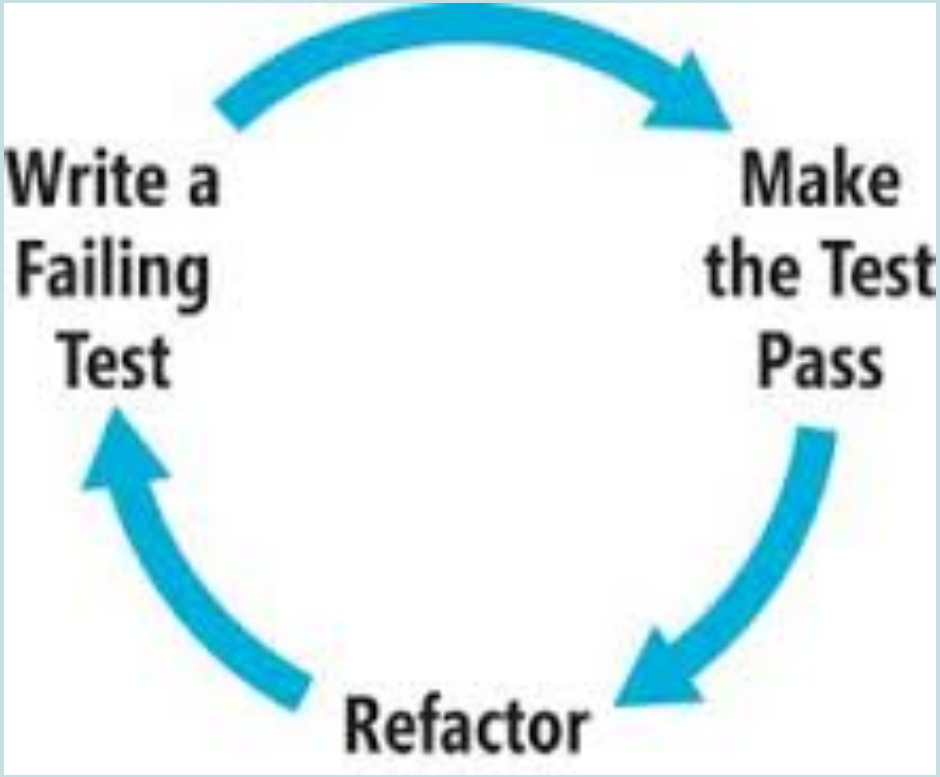
1. Open Xcode
2. Make sure Xcode is 8.x
3. Open Main.storyboard
4. Search for TextView in the Object Library
5. Drag over to Storyboard
6. Run

Lab 2 - Hello World

1. Open Terminal
2. cd /path/to/HelloWorld
3. Run the command

```
xcodebuild build -project HelloWorld.xcodeproj
    -scheme HelloWorld
    -destination 'platform=iOS Simulator,name=iPhone 7 Plus'
```

Test Driven Development



Test Driven Development

- Write test first
- See it fail
- Write simplest possible solution to get test to pass
- Refactor
- Wash, Rinse, Repeat

Uncle Bob's Rules

- You are not allowed to write any production code unless it is to pass a failing unit test
- You are not allowed to write any more of a unit test than is sufficient to fail, and compilation failures are failures
- You are not allowed to write any more production code than is sufficient to pass the one failing unit test

Test Driven Development

- Built in regression testing
- Longer life for your codebase
- YAGNI feature development
- Red/Green/Refactor helps kill procrastination

TDD

You can't TDD w/o unit testing

TDD means writing the tests before the code

TDD is more painless than classic unit testing

Unit Testing

You can unit test w/o TDD

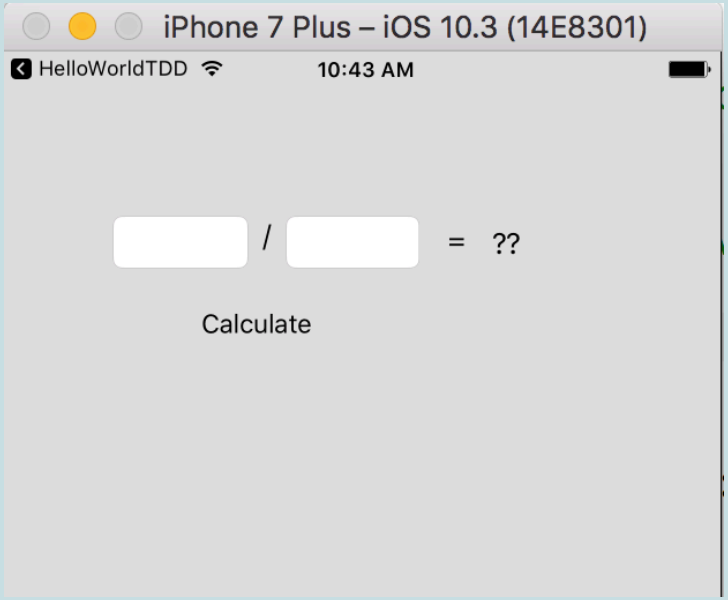
Unit tests don't mandate when you write the tests

Unit tests are often written at the end of a coding cycle

Lab 3 - Hello World TDD

Features

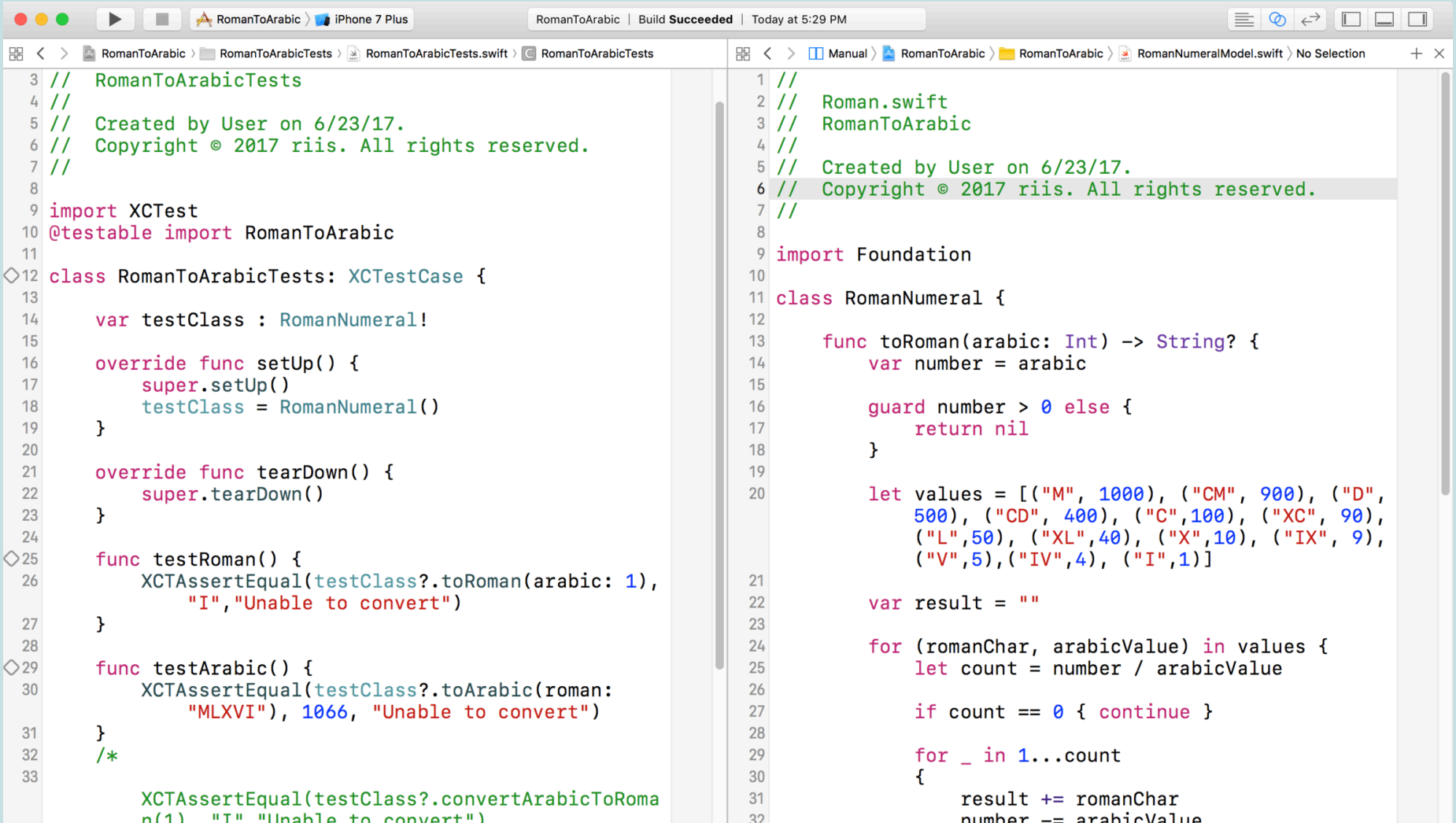
- User gets valid result when dividing divisor by dividend
- User can not divide by zero



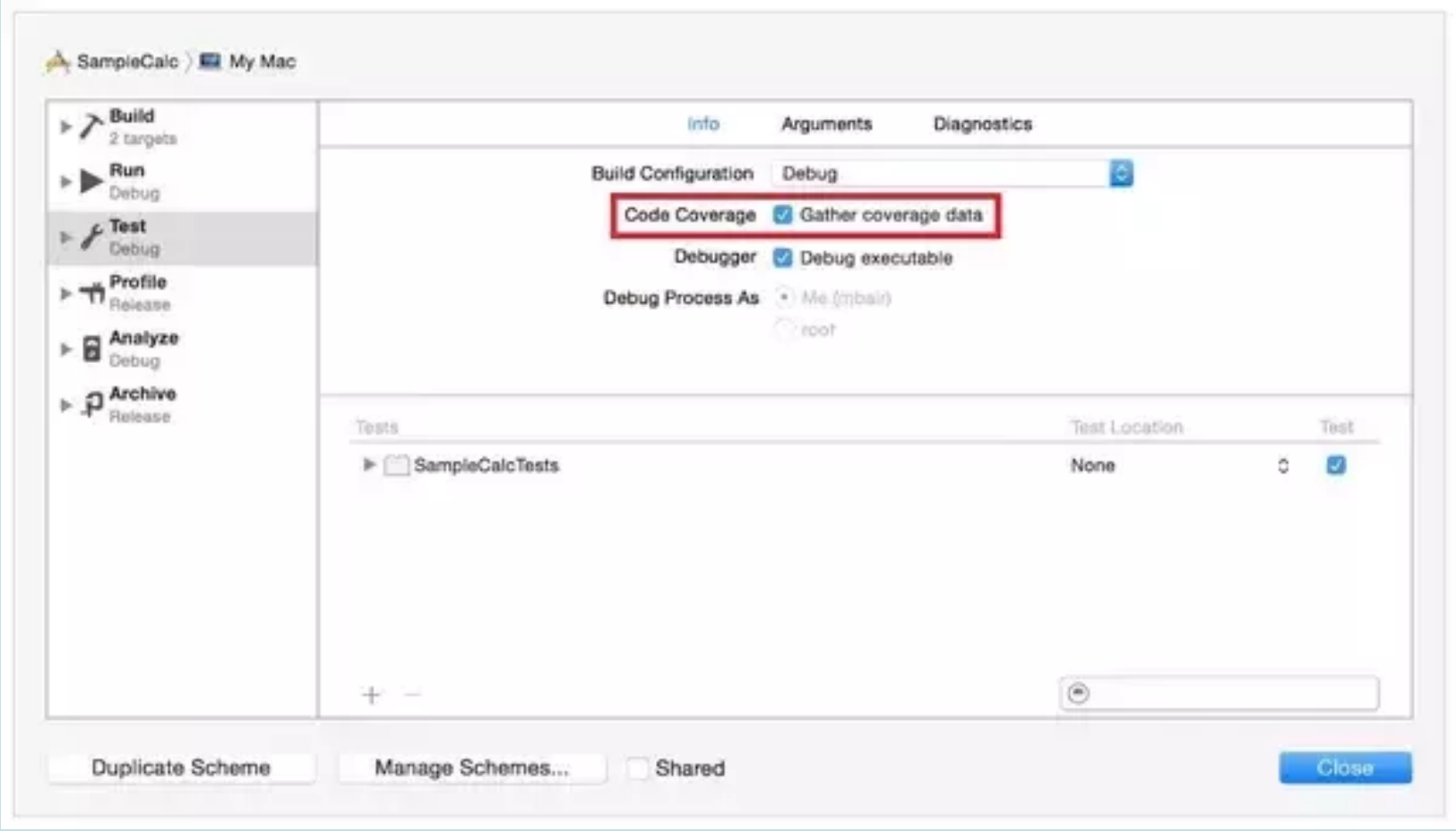
Lab 3 - Hello World TDD

1. Create Xcode project called Division
2. Add two text fields, divide and results label and button
3. Create IBoutlets, IBActions
4. Create Calculator object
5. Create Unit test for feature 1
6. Run test - test will fail
7. Fix Calculator code so test passes
8. Create Unit test for feature 2
9. Run test - test will fail
10. Fix Calculator code so test passes
11. Run test
12. Refactor - fix ViewController code
13. Turn on code coverage
14. View code coverage

XCode TDD


RomanToArabic 〉 iPhone 7 Plus
RomanToArabic | Build Succeeded | Today at 5:29 PM
RomanToArabic 〉 RomanToArabicTests 〉 RomanToArabicTests.swift 〉 RomanToArabicTests
Manual 〉 RomanToArabic 〉 RomanToArabic 〉 RomanNumeralModel.swift 〉 No Selection
3 // RomanToArabicTests
4 //
5 // Created by User on 6/23/17.
6 // Copyright © 2017 riis. All rights reserved.
7 //
8
9 import XCTest
10 @testable import RomanToArabic
11
12 class RomanToArabicTests: XCTestCase {
13
14 var testClass : RomanNumeral!
15
16 override func setUp() {
17 super.setUp()
18 testClass = RomanNumeral()
19 }
20
21 override func tearDown() {
22 super.tearDown()
23 }
24
25 func testRoman() {
26 XCTAssertEqual(testClass?.toRoman(arabic: 1),
"I","Unable to convert")
27 }
28
29 func testArabic() {
30 XCTAssertEqual(testClass?.toArabic(roman:
"MLXVI"), 1066, "Unable to convert")
31 }
32 /*
33
XCTAssertEqual(testClass?.convertArabicToRoma
1 //
2 // Roman.swift
3 // RomanToArabic
4 //
5 // Created by User on 6/23/17.
6 // Copyright © 2017 riis. All rights reserved.
7 //
8
9 import Foundation
10
11 class RomanNumeral {
12
13 func toRoman(arabic: Int) -> String? {
14 var number = arabic
15
16 guard number > 0 else {
17 return nil
18 }
19
20 let values = [("M", 1000), ("CM", 900), ("D",
500), ("CD", 400), ("C",100), ("XC", 90),
("L",50), ("XL",40), ("X",10), ("IX", 9),
("V",5),("IV",4), ("I",1)]
21
22 var result = ""
23
24 for (romanChar, arabicValue) in values {
25 let count = number / arabicValue
26
27 if count == 0 { continue }
28
29 for _ in 1...count
30 {
31 result += romanChar

XCode TDD


SampleCalc
My Mac
Build
2 targets
Run
Debug
Test
Debug
Profile
Release
Analyze
Debug
Archive
Release
Info
Arguments
Diagnostics
Build Configuration
Debug
Code Coverage
Gather coverage data
Debugger
Debug executable
Debug Process As
Tests
Test Location
Test
SampleCalcTests
None
Duplicate Scheme
Manage Schemes...
Shared
Close

Exercism Kata

EXERCISM Languages Donate Log in with GitHub

HOME ▸ Languages ▸ Swift ▸ Roman Numerals

Roman Numerals in Swift

Write a function to convert from normal numbers to Roman Numerals.

```
1  exercism fetch swift roman-numerals
```

Readme | Test Suite

Roman Numerals

Write a function to convert from normal numbers to Roman Numerals.

The Romans were a clever bunch. They conquered most of Europe and ruled it for hundreds of years. They invented concrete and straight roads and even bikinis. One thing they never discovered though was the number zero. This made writing and dating extensive histories of their exploits slightly more challenging, but the system of numbers they came up with is still in use today. For example the BBC uses Roman numerals to date their programmes.

The Romans wrote numbers using letters - I, V, X, L, C, D, M. (notice these letters have lots of straight lines and are hence easy to hack into stone tablets).

```
1   1  => I
2  10  => X
3   7  => VII
```

There is no need to be able to convert numbers larger than about 3000. (The Romans themselves didn't tend to go any higher)

Wikipedia says: Modern Roman numerals ... are written by expressing each digit separately starting with the left most digit and skipping any digit with a value of zero.

To see this in practice, consider the example of 1990.

In Roman numerals 1990 is MCMXC:

1000=M 900=CM 90=XC

Exercism Kata

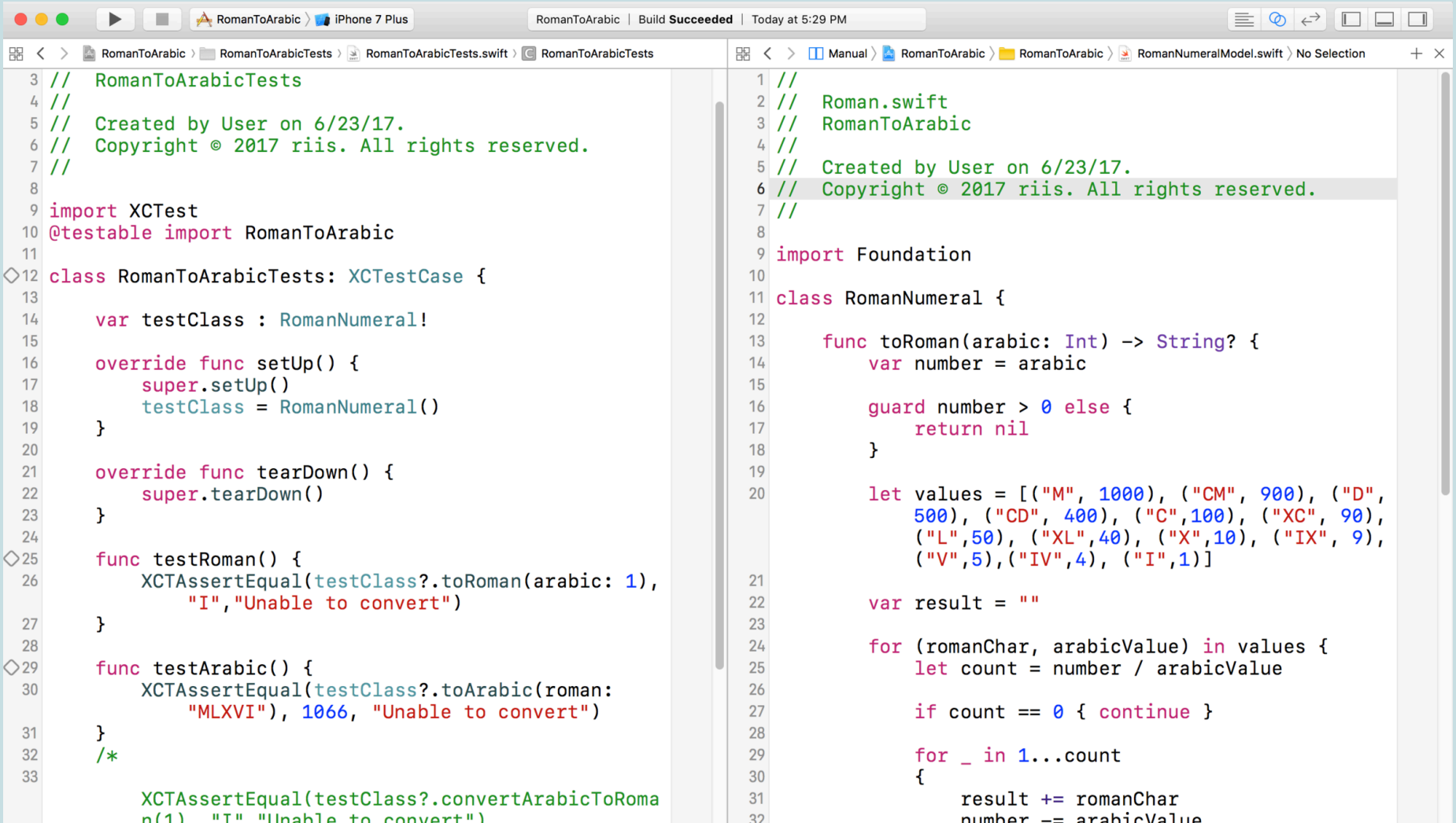
Write a function to convert from Arabic numbers to Roman Numerals

Write a function to convert from Roman Numerals to Arabic numbers

Exercism Kata

1. Create ArabicToRoman project in Xcode
2. Check Unit Tests
3. Open ArabicToRomanTests - use the two screens in Xcode
4. Create ArabicToRomanModel.swift
5. Create toRoman stub function
6. Create testConvertToRoman test
7. Run test from command line, test will fail
8. Fix toRoman stub function
9. Run test from command line, test passes
10. Refactor
11. Repeat for second feature

XCode TDD


RomanToArabic 〉 iPhone 7 Plus
RomanToArabic | Build Succeeded | Today at 5:29 PM
RomanToArabic 〉 RomanToArabicTests 〉 RomanToArabicTests.swift 〉 RomanToArabicTests
Manual 〉 RomanToArabic 〉 RomanToArabic 〉 RomanNumeralModel.swift 〉 No Selection
3 // RomanToArabicTests
4 //
5 // Created by User on 6/23/17.
6 // Copyright © 2017 riis. All rights reserved.
7 //
8
9 import XCTest
10 @testable import RomanToArabic
11
12 class RomanToArabicTests: XCTestCase {
13
14 var testClass : RomanNumeral!
15
16 override func setUp() {
17 super.setUp()
18 testClass = RomanNumeral()
19 }
20
21 override func tearDown() {
22 super.tearDown()
23 }
24
25 func testRoman() {
26 XCTAssertEqual(testClass?.toRoman(arabic: 1),
"I","Unable to convert")
27 }
28
29 func testArabic() {
30 XCTAssertEqual(testClass?.toArabic(roman:
"MLXVI"), 1066, "Unable to convert")
31 }
32 /*
33
XCTAssertEqual(testClass?.convertArabicToRoma
1 //
2 // Roman.swift
3 // RomanToArabic
4 //
5 // Created by User on 6/23/17.
6 // Copyright © 2017 riis. All rights reserved.
7 //
8
9 import Foundation
10
11 class RomanNumeral {
12
13 func toRoman(arabic: Int) -> String? {
14 var number = arabic
15
16 guard number > 0 else {
17 return nil
18 }
19
20 let values = [("M", 1000), ("CM", 900), ("D",
500), ("CD", 400), ("C",100), ("XC", 90),
("L",50), ("XL",40), ("X",10), ("IX", 9),
("V",5),("IV",4), ("I",1)]
21
22 var result = ""
23
24 for (romanChar, arabicValue) in values {
25 let count = number / arabicValue
26
27 if count == 0 { continue }
28
29 for _ in 1...count
30 {
31 result += romanChar

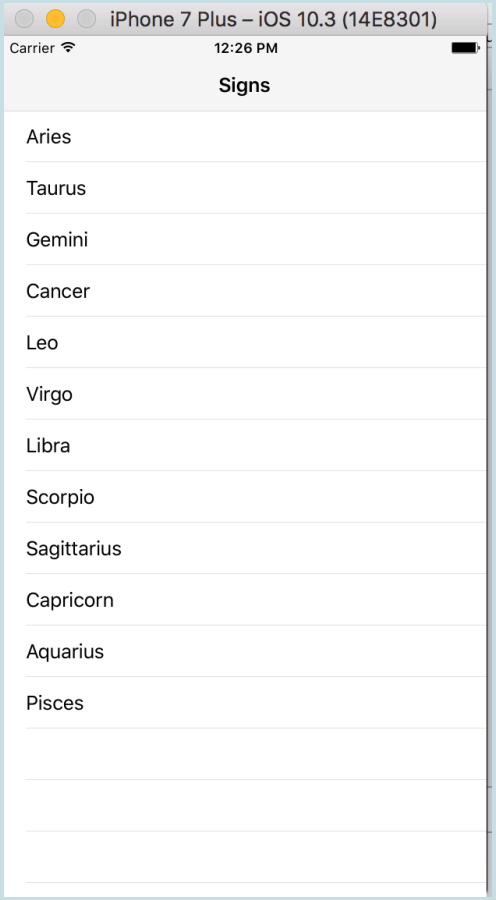
Sample App

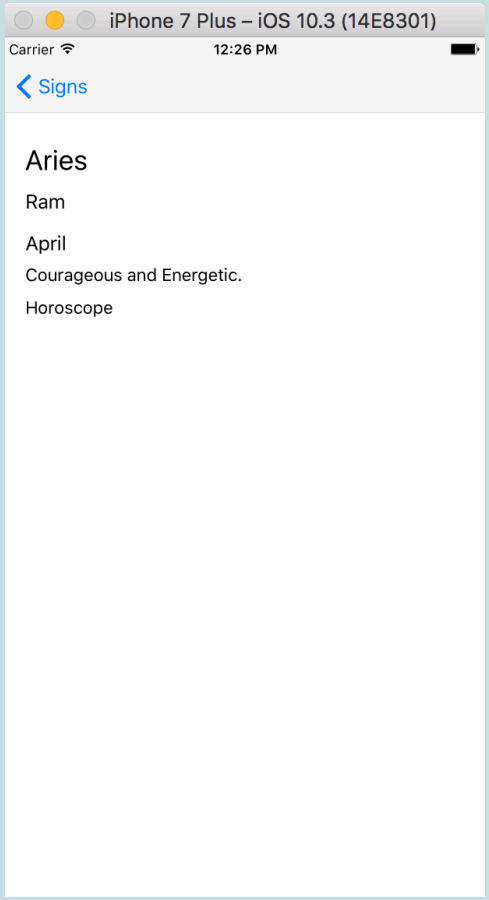
Feature 1 - Display list of Zodiac Signs

Feature 2 - Display descriptions of Sign

Feature 3 - Display daily horoscope

Sample App


iPhone 7 Plus – iOS 10.3 (14E8301)
Carrier
12:26 PM
Signs
Aries
Taurus
Gemini
Cancer
Leo
Virgo
Libra
Scorpio
Sagittarius
Capricorn
Aquarius
Pisces


iPhone 7 Plus – iOS 10.3 (14E8301)
Carrier
12:26 PM
Signs
Aries
Ram
April
Courageous and Energetic.
Horoscope

Resource

https://medium.com/@ynzc/getting-started-with-tdd-in-swift-2fab3e07204b

https://github.com/samantha-wong/FizzBuzz

https://gist.github.com/kumo/472043819fec7a3737f8

https://github.com/pigmonchu/TDDiOS

http://rshankar.com/test-driven-development-in-swift/