

CSC 371 Mobile Application Development for iOS

Programming Assignment 2 Swift Programming

Due Date & Submission

- Assignment due on Thursday January 19, 2017, 11:59pm
- Submit your assignment in D2L Dropbox.
- Submit a single zip file that contains the contents of the project folder
 - To zip your project folder: Ctrl-click your project folder and select “Compress ...” from the context menu.
- **It is mandatory to use Xcode 8 and Swift 3 for this assignment.**
- Include only your source code files, including
 - *.swift, *.plist, *.xib, *.storyboard
 - image files
 - project files (.xcodeproj)
 - test folders
- You must use a unique prefix for the project name. (I suggest you use your last name and first initial as your prefix.) **Please use the same prefix for all your assignments.**
 - Note you only need to use the prefix for the project name. It is not necessary to use the prefix on other files in your project.
- Do not include unused or unrelated files.
- Before you submit, build and run the project, make sure everything compiles and works. Close your project before zipping the folder.
- Here are the most common reasons assignments are marked down:
 - Project does not build.
 - Project does not build without warnings.
 - One or more items in the Requirements section were not satisfied.
 - A fundamental concept was not understood.
 - Code is sloppy and hard to read (e.g. indentation is not consistent, etc.).
 - Your solution is difficult (or impossible) for someone reading the code to understand due to lack of comments, poor variable/method names, poor solution structure, etc.

Goals

- Explore the Swift programming language
- Write a simple function and a class in Swift
- Experiment with Playground

Assignments

1. Create a new Playground. Include all the code of this assignment in a single Playground. Compress the Playground to a zipped archive file. Submit the zip file in D2L
2. Write a function in Swift that performs the following tasks
 - a. Function rank performs a binary search of a key in a sorted integer array. If the key appears in the array, the rank function returns the index of key in the array. If the key does not appear in the array, the rank function returns -1. (A Java implementation of the rank function can be found in *Algorithm*, 4th ed. by R. Sedgewick and K. Wayne, the textbook of the data structures course.)
 - b. Create a sorted integer array to test the rank function. Make several calls to the rank function. Verify that the results are correct.
3. Write a more complete version of the Fraction class we have discussed in the lecture. **Start from scratch and type in all the code yourself.** (You will learn more from the mistakes you make along the way.)
 - a. Add methods to the version 4 (Program 5.6 in *A Swift Primer, Part 1*) of the Fraction class to compute the subtraction, multiplication, and division of two fractions in a similar way to the add method (for the formulae to compute the subtraction, multiplication, and division of two fractions, see http://en.wikipedia.org/wiki/Rational_number). Each method should take another fraction object as the parameter and return the result as a new Fraction object. The resulting fraction should not have common factors in its numerator and denominator.
 - b. Write test code to test each of the new methods of the class with different input values. Verify that your results are correct.