

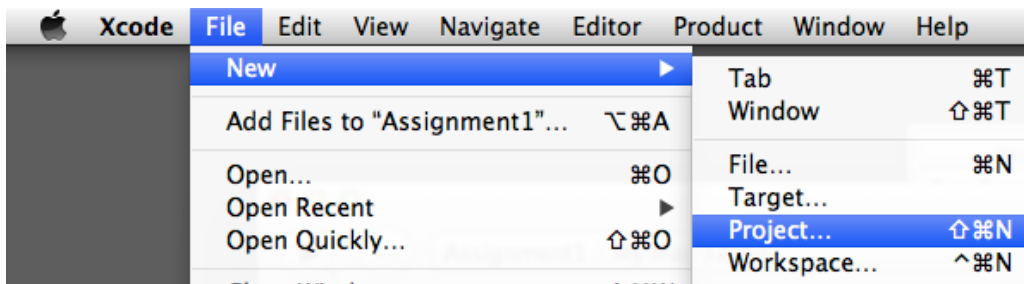
CS 646 iPad/iPhone Application Development  
Fall Semester, 2013  
Assignment 1  
© 2013, All Rights Reserved, SDSU & Roger Whitney  
San Diego State University -- This page last updated 8/28/13

Assignment 1 (50 points)  
Due Sept 4 11:59 pm  
Version 1.1 (8/28/13)  
Objectives

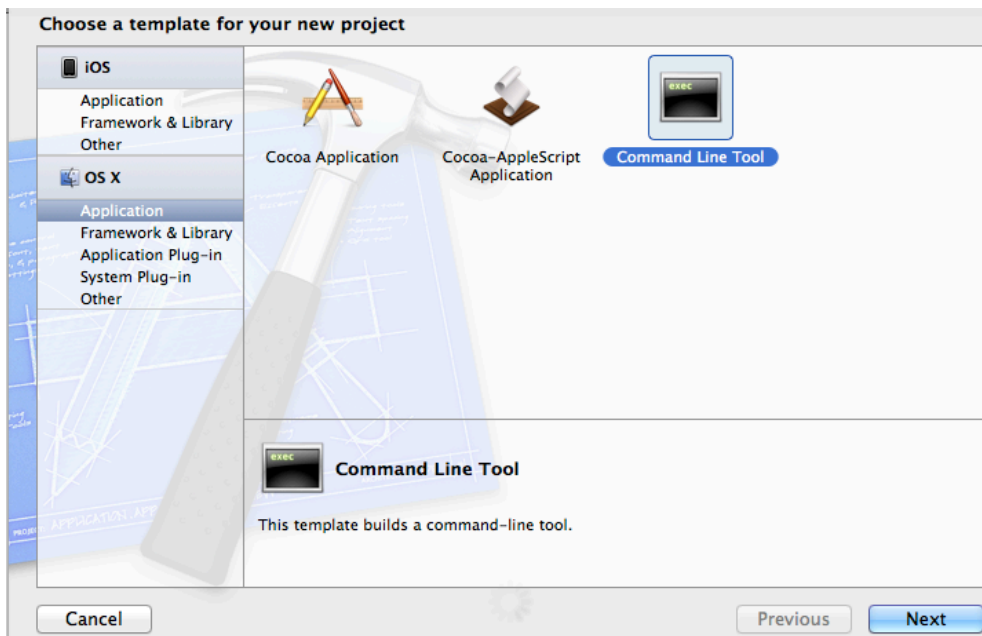
1. Get started using Xcode.
2. Learn basic Xcode operations.
3. Learn how to use Xcode documentation for iOS API.

Programming Problem (5 points each)

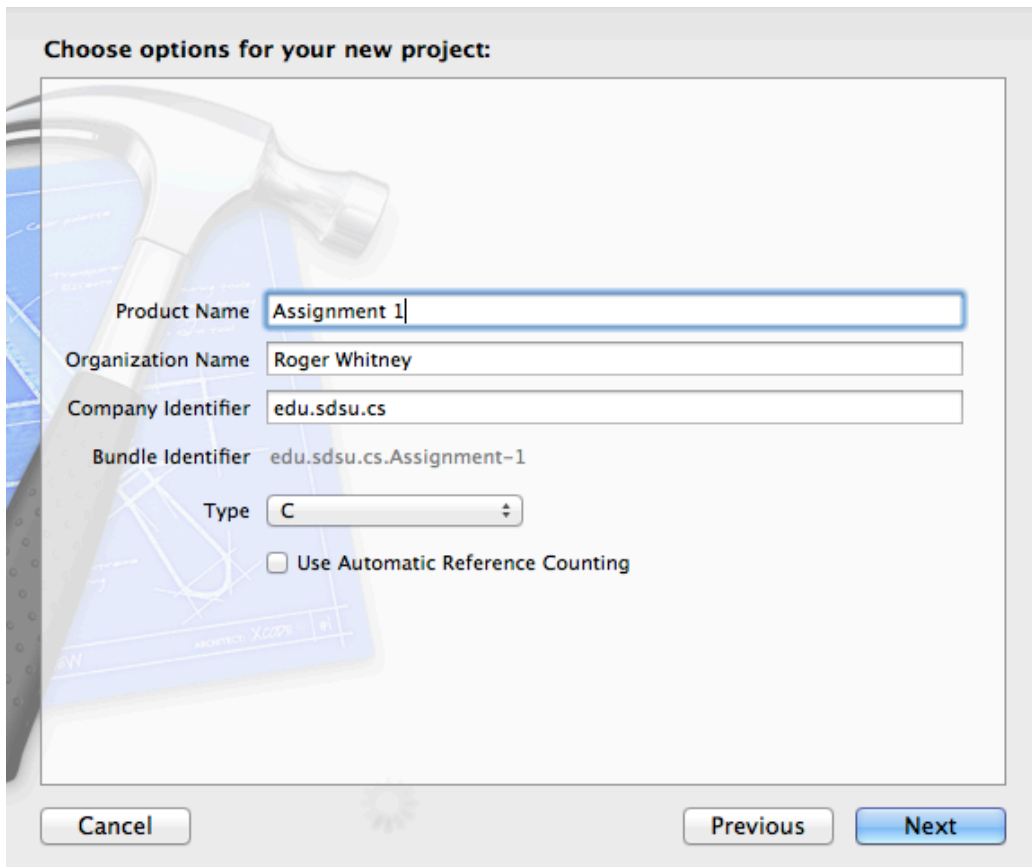
One of the main goals of this assignment to start getting familiar with Xcode. So you must use Xcode for this assignment. You will be turning in an Xcode project. In Xcode create a new project. To do this in the "File" menu select "New" then Project".



Then under OS X select "Application" then "Command Line Tool". Click on the "Next" button.



Fill in the next window with your information.

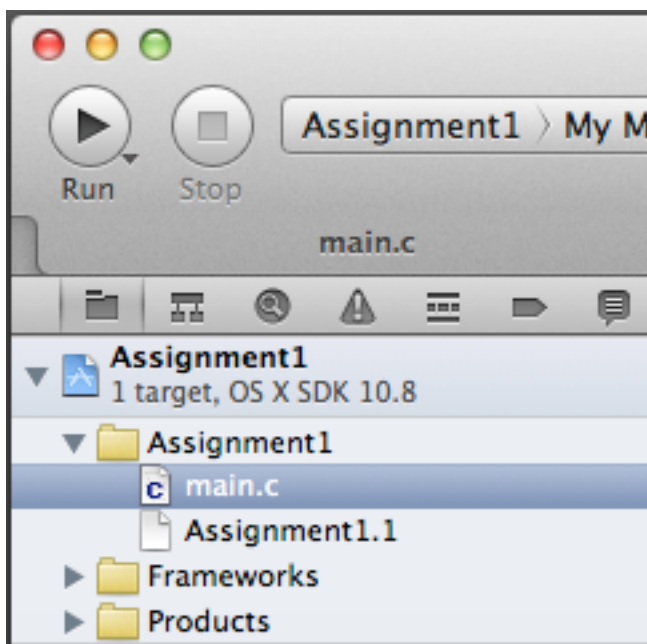


The image shows the 'Choose options for your new project' dialog box in Xcode. The background features a faint image of a hammer and a blueprint. The dialog contains the following fields and options:

- Product Name:** Assignment 1
- Organization Name:** Roger Whitney
- Company Identifier:** edu.sdsu.cs
- Bundle Identifier:** edu.sdsu.cs.Assignment-1
- Type:** C
- ☐ Use Automatic Reference Counting

At the bottom, there are three buttons: 'Cancel', 'Previous', and 'Next'.

After clicking "Next" save your project. Once the project is created find the "main.c" file in the left most pane in Xcode. Click on the file to open it. Add your code to that file.



Now the programming questions.

1. Write a c function, call it cubeTable, that has one argument, an integer. Lets call it N. The function prints out on the console the values k and  $k^3$  for the values  $k = 1, 2, \dots, N$ . Print each value of k on a separate line.
2. Add a breakpoint in the cubeTable method. Step through the cubeTable method until  $k = 3$ . In the debugger change the value of N to -1. What happens when you continue to run the program?
3. Write a c function, called runningSum, that has one int argument and returns an int. The return value is the sum of the current and past values of the arguments passed to the function. The history of the arguments restarts at zero each time your program starts. Test your function by calling it several times in main and printing the return value each time. Do not use global variables (think static).

```
//Start of program
x1 = runningSum(2);
// x1 == 2
x2 = runningSum(2);
// x2 == 4
x3 = runningSum(3);
// x3 == 7
x4 = runningSum(5);
// x4 == 12
```

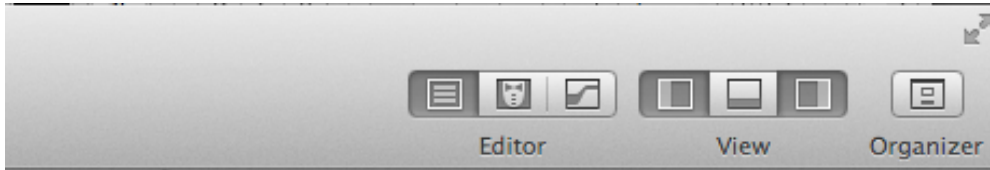
4. Add a break point at the start of your runningSum program. Have the break point trigger for the first time the third time runningSum is called. Change the value of your static variable to 2 then continue running the program. When you do this what will be the value of x4 in the above code?
5. The Fibonacci numbers are defined as:  $F_n = F_{n-1} + F_{n-2}$  where  $F_1 = F_2 = 1$  and n is a positive integer. Write a recursive c function, called fibonacci, that with input n returns the nth Fibonacci number  $F_n$ .
6. Profile compute(10000) where compute is given below. Use the Time Profiler. (Select "Profile" in the "Product" menu. Then in the dialog window select "CPU" under "Mac OS X", then in the right pane select "Time Profiler" and click on the "Profile" button. In the profile results what percentage of time did the sin function take?

```
float compute(int n) {
    float y = 1.1;
    for (int k = 0; k < n; k++)
        for (int j = 0; j < n; j++)
            y = sin(k*j + y);
    return y;
}
```

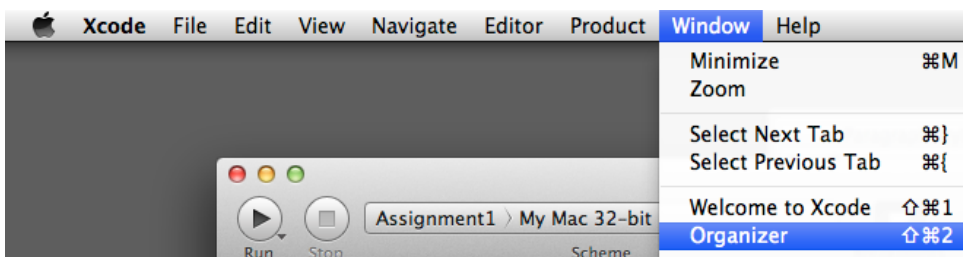
## Questions (5 points each)

No doubt that you can find the answers to these questions using a Google search, but don't. The entire point of these questions is to get you used to using the documentation for iOS included with Xcode. So use Organizer in Xcode to find the answers.

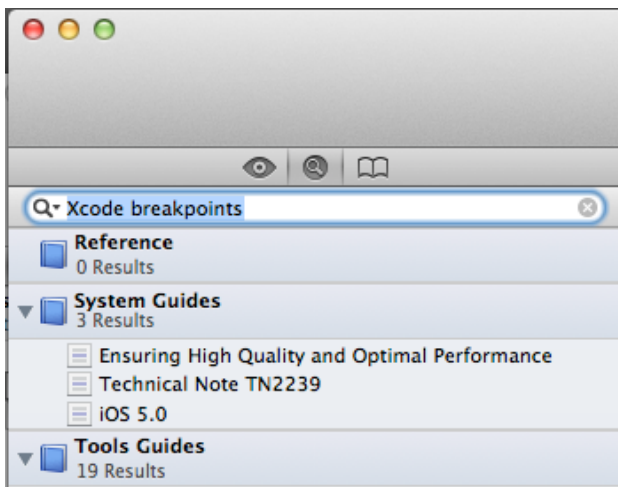
See below on how to create a project in Xcode. Once you have an Xcode project you can open the "Organizer" in Xcode by clicking on the "Organizer" button in the upper right of the Xcode window.



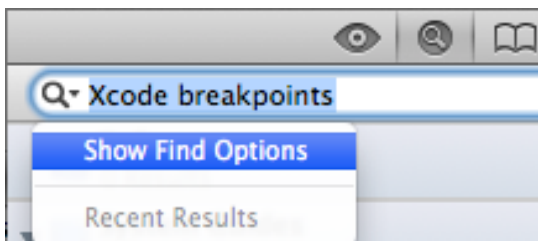
Or by the "Organizer" item in the "Window" menu.



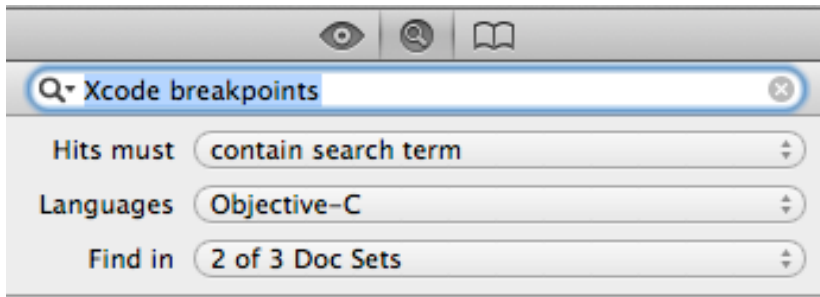
In the left pane of the "Organizer" there is a search box.



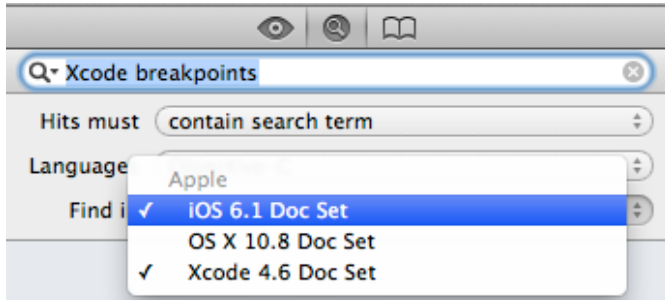
You can set the find options by clicking on the magnifying glass.



There are three options you can set.



You get better result by selecting only "Objective-C" as the language. In the "Find In" you get faster results and fewer repeats if you just select one Xcode doc set and one iOS doc set.



Now finally the questions.

7. What version of iOS added the method **description** to the class Object?
8. What method should one use instead of **+ stringWithContentsOfURL:** in NSString? Why is stringWithContentsOfURL: deprecated?
9. What is the difference between the methods **valueForKey:** and **objectForKey:** in the NSDictionary class?
10. What protocols does the NSArray class conform to?

#### What to Turn in

Create a Xcode project for the assignment. In your project add a file called "Answers.txt". In this file put the answers to the "Questions" 7-10 and "Programming Problems" 2, 4, 6. Xcode places the project in its own directory. Place the directory (and all its contents) into a zip file. Turn in your zipped file using assignment 1 link on blackboard.

#### Late Penalty

An assignment turned in 1-7 days late, will lose 3% of the total value of the assignment per day late. The eight day late the penalty will be 40% of the assignment, the ninth day late the penalty will be 60%, after the ninth day late the penalty will be 90%. Once a solution to an assignment has been posted or discussed in class, the assignment will no longer be accepted. Late penalties are always rounded up to the next integer value.

## Modifications

### 1.1 Corrected initial seeds for Fibonacci sequences.