Overview

During this lab, you are provided skeleton code and tasked to create a calculator.

- Check out the skeleton code from the Subversion repository
- Edit the model so the calculator passes the given tests
- Expand the unit tests

Submission

Lab evaluation form
Online feedback form

Note: If you are unable to complete the lab during your scheduled lab hours, please be ready to demonstrate it at the beginning of the next lab. iPods are available for checkout from CSG. Labs are considered late (20% penalty) if they are not turned in by the end of the next lab period.

Resources

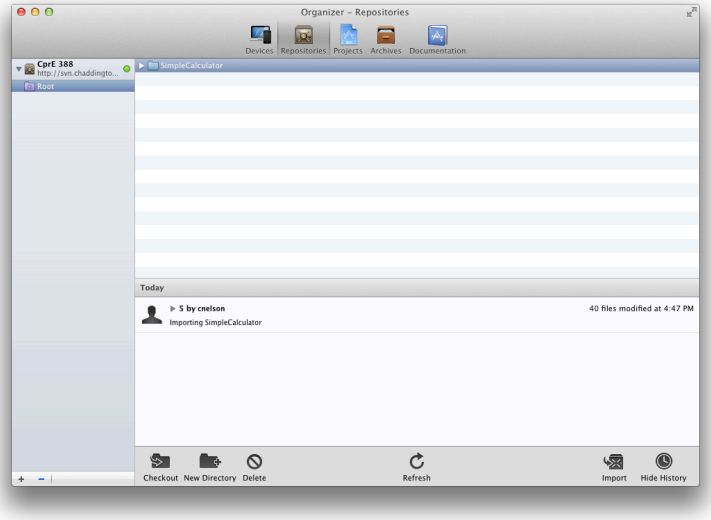
String Format Specifiers

Calculator

One of the many benefits of the Model-View-Controller software architecture is the ability to separately test your model and your visual display. In this lab, you will implement the Model part of a calculator app. The View and Controller parts of the application have already been implemented, and you will be graded based on your implementation of the model (Calculator.h and Calculator.m).

Step 1) Check out the Calculator project from the Subversion repository at:
http://svn.chaddington.com/cpre388

- Open Xcode
- File -> Source Control -> Repositories
- Click the "+" in the lower left corner to add a repository.
- Follow the prompts.
- Select the "SimpleCalculator" project in the root of the repository, and select "Check Out"
- When prompted, open the project with Xcode.



Step 2) You will be working entirely in Calculator.h and Calculator.m in order to implement the calculator. Complete the functions inside Calculator.m, giving the calculator functionality similar to MS Calculator (operations accumulate).

Step 3) Run the sample test cases on your app (Project -> Test). These tests should pass. Test your project in the simulator as well. **You do not need to worry about all edge cases (lucky you!).** While you will be awarded a few points for expanding the test cases, and the majority of the points are awarded for implementing a working calculator.

Question: How to I convert between NSString and primitive types?

Short Answer: Did you read the link given in the resources above?

Long Answer:

```
// Simple conversion
double number1 = 12.3;
NSString *string1 = [NSString stringWithFormat:@"%g", number1];

NSString *string2 = @"12.3";
double number2 = [string2 doubleValue];
```