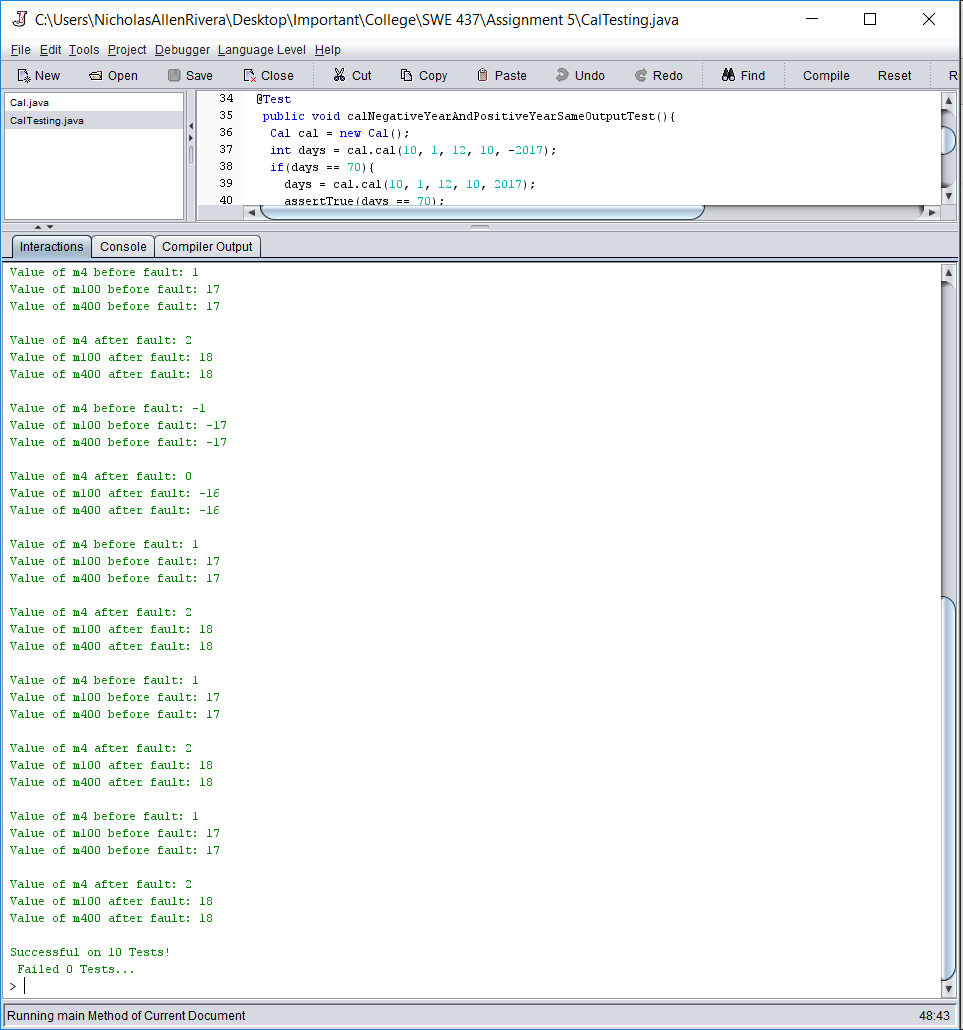
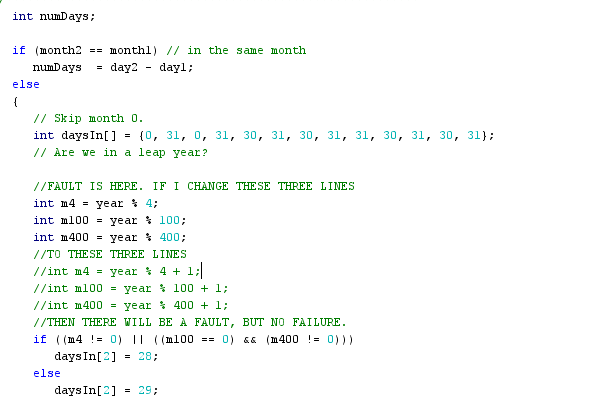
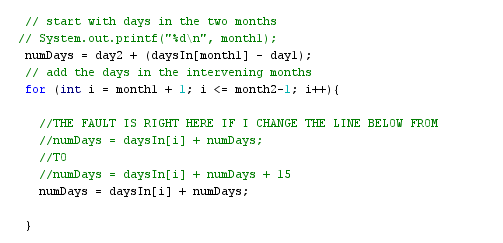
Assignment 5:

I created my tests mostly around checking if the method in which to retrieve the number of days between two dates is correct. To elaborate on this I only had one exception test calIndexOutOfBoundsExceptionTest() because I couldn’t find as many opportunities for the code to fail in that regard as I could find it to fail in regards to finding the correct number of days. Most of the times the algorithm reported a correct value without there being any kind of fault, but there were some problems. The biggest one I could find was the leap year functionality, most of the possible incorrect input is stemmed in the main method of the cal class, but not for leap years. The decision for a year to be considered a leap year lies within the cal method, so this allows for errors to exist. My test calNonLeapYear29DaysTest()’s purpose was to find if I had a non-leap year such as 2017, and while on the month of February, input 29 as the value of day1 or day2 as if February, 2017 could have a 29th day. This test did not return an error indicating to me that incorrect input can get past all the other methods before reaching the cal method. I also tested using negative numbers for certain parameters, I checked if the year regardless of negativity would return the same value, and I checked if the day would return a different number if negative, and they both came back with the correct responses. I also checked if inputting a later date and then an earlier date would create a negative result, but this test succeeded as well. In conclusion these tests shed a lot of light about the functionality of this program, and I even discovered some pre-existing faults within the code.

I created some faults to test the ability of this program, and I learned that this program doesn’t have many guards against processing leap years incorrectly.

My first fault, which my tests do not find, changes what it means to be a leap year by incrementing m4, m400, and m100 by 1 which affects the proceeding if statement as shown below. Code can still execute without failure even if the leap years are completely wrong. 

My second fault, which my tests do find, adds 15 to numDays which affects the return value of three of my tests as shown below.



Changing this line inevitably results in below. The three failed tests were the only tests which executed the for loop causing them to fail. 