**Program Description**

Program retrieves date data and can do various calculations on it such as days in month, add and subtract time from date, calculate the time between two dates in time, check if a date is valid, and check if a year is a leap year.

The SLOC is 160 lines long and has 8 public methods.

One bug was that the current date was wrong and the calculation was adding days ended up displaying wrong as well. For example adding 20 days can change the year by -+ 10. Which might have to do with the flip servers, but I am not completely sure. The code can be found in the first public function addtoDate() on line 13.

**Utility Tool**

For the assignment, I used evosuite with mvn command line. Evosuite works by systematically working through the functions and methods of a program and automatically generate Junit tests to challenge them.

Over all after compiling and generating the unit tests, which 13 where created in the test suite, it was nice to be able to look through it and see the potential it can bring to my own projects. The hardest part of the assignment was finding something online that would work well, and I don’t think the program I found challenged Evosuites generation, but overall the tests seemed to work well. When running “mvn test” it was also nice to see the output of the tests. I think the biggest complaint I have about it was setting it up and learning the basics of maven and POM’s dependency requirements. I would like to see a more automatic way to install Evosuites and maybe a GUI interface tool.

**Test Cases**

13 Test cases where made and the tests all passed, but did display that there where calculation errors in the dates, overall the problem can be addressed by looking at the output of the test cases and manually correcting the code.

In this case I didn’t introduce bugs since there seemed to being existing problems when I ran the program.

Link http://www.wideskills.com/java-tutorial/java-date-utility