WRITE-UP

Justification of why your JUnit tests are appropriate:

As a rundown of some of my JUnit tests, I tested both the number of days in a normal year and the number of days in leap year to make sure that the calculation between the first day of the year and last day of the year was +1 from that in a normal year. As a result, the tests passed correctly with a normal year consisting of 364 days in between, and a leap year consisting of 365 days in between. Another test I did to make sure the program ran as it should was to calculate the days between the same day. So given the same month and day for the parameters month1/month2 and day1/day2, it should output 0 as there are no days in between, which it accurately depicted. Furthermore, I did a test to check if the program would count a month that does not have 31 days. To do this, I calculated the days between June 31st, and September 31st, both months that don't have a 31st. As a result, it correctly did not count them as days in the calculation. Similarly, I made 2 tests that resembled each other with the one difference being the year. First, I made a test that calculated the days between February 29th and March 8th in 2016 (a leap year). Correctly calculated, the result is 8 days as February 29th is counted. Then, I made the same test and calculated the days between the same days as mentioned before, but I changed the year to 2015(a non-leap year). As expected, it did not count February 29th because it is not a day in a regular year. All of these JUnit tests and more were exemplary in demonstrating how the program correctly tests the number of days between two given days in the same year.

Discussion of fault missed by your test set:

A fault missed by 2 of my test sets were when I passed in February 30th and 31st into the parameters as one of the two given days. In my first test, I calculated the days between February 17th and **February 30th** in **2007** (a non-leap year). I set *assertEquals* to 11, as that is how many days are between February 17th and February 28th, thinking that it should not be counting the 29th (as it's not a leap year), nor the 30th (because it's not an actual day). To my surprise though, I got that my test failed, stating "expected <13> but was: <11>." This made me realize that it was counting the 29th and 30th as dates in February, even though that was not possible for that year or at all. To further investigate, I created another similar test case calculating the duration between February 17th and **February 31st in 2008** (a leap year). I set *assertEquals* to 12 this time, considering the 29th would count since it was a leap year. Once again, though, my test failed, stating "expected <14> but was: <12>. It incorrectly counted up till the 31st, which is a fault.

Discussion of fault found by your test set:

A fault found by my test set was when I set the parameters to have a later day in April come first in the parameter, followed by an earlier date in April. Why? Because in the Cal program, it states that it will calculate the number of days between two given days by doing "day2-day1." The parameter is set to be in the format (int month1, int day1, int month2, int day2, int year). What I tested was "(04, 17, 04, 10, 2009)." I wanted to test this to see if a negative number of days would be output. This test was essentially doing day2(10) – day1(17). As predicted, a negative number of days was output (-7). This is not a possible number of days; the result should only be positive. Thus, this was a fault found by my test.