Homework 5: how to figure out what you missed

Get Grade5.java. Compile it together with your code.

The Grade5 program creates a series of Moment objects, and then takes two integer inputs that specify which test should be carried out on which of these objects.

Some of you did not create constructors that met the specifications given in the `MomentInterface.java` file. In this case, your program might not compile or work correctly. Nissa cut you some slack and fixed your constructors before running the tests. You may have to do this also to get the test to run.

To run Test 1.2, run Grade5, and type in the inputs 1 and 2. To run Test 2.3, tun Grade5 and type in 2 and 3, etc. (The program doesn't prompt for them.) In general, for Test i.j, input i and j.

These two inputs are assigned to `choice1` and `choice2` in the program. To see what test is carried out when you type in these inputs, look in the code to see which statements get executed when `choice1` and `choice2` have the values you gave as your inputs.

Below is the number of points assigned to each test, as well as the correct answer your program should output.

Many of you missed a large number of problems because, for example, you didn't get your `getMilitaryHour()` working correctly, but made calls to this broken method inside other methods. You need to test all of your methods, and, if they are broken, tracked down the problem before turning in the program. (See the guidelines about testing given in the handout for the homework.)

```
Test 1.1 - get default hour (2pts)12
Test 1.2 - get am hour (2pts)9
Test 1.3 - get pm hour (2pts)
4
Test 2.2 - military hour <12 (1pt)9
Test 2.3 - military hour =12 (1pt)12
Test 2.4 - military hour >12 (1pt)
16
Test 2.5 - military hour =0 (1pt)0
Test 3.1 - get def minute (2pts)0
Test 3.2 - get am minute (2pts)34

Test 4.1 - get def second (2pts)0
Test 4.2 - get am second (2pts)32
Test 5.1 - get def pm/am (1pt)false

Test 5.2 - get am pm/am (1pt)false
Test 5.3 - get noon pm/am (1pt)true
Test 5.4 - get pm pm/am (1pt)true

Test 6.1 - def isLegal (1pt)false
Test 6.2 - 2/29/2500 isLegal (2pts)false
```

```
Test 6.3 - second = 60 isLegal (1pt)false

Test 6.4 - hour = 24 isLegal (1pt)false
Test 6.5 - legal isLegal (4pts)true
Test 7.1 - secondsUntil same date (1pt)-29

Test 7.2 - secondsUntil, first < second (1pt)30675630
Test 7.3 - secondsUntil, first > second (1pt)-31535722
Test 7.4 - first date illegal (2pts)
Test 8.1 - minutesUntil same date (1pt)9
Test 8.2 - minutesUntil, first < second (1pt)87929
Test 8.3 - minutesUntil, first > second (1pt)
-89252
Test 8.4 - minutesUntil, 30 second diff (1pt)525541
Test 9.1 - hoursUntil same date (1pt)5
Test 9.2 - hoursUntil, first < second (1pt)
13126
Test 9.3 - hoursUntil, first > second (1pt)-4326
Test 9.4 - hoursUntil, 30 min diff (1pt)11629
Test 9.5 - second date illegal (2pts)
Test 10.1 - equal dates compareTo (4pts)0
Test 10.2 - compareTo first < second by seconds (1pt)-1
Test 10.3 - comapreTo first > second by minutes (1pt)
1
Test 10.4 - compareTo first > second by hours (1pt)1
Test 10.5 - compareTo same date, AM vs PM (1pt)-1
Test 11.1 - def toString (1 works - 1 perfect)
0/0/0 12:0:0 AM
Test 11.2 - am toString (1 works - 1 perfect)1/1/1943 9:34:32 AM
Test 11.3 - pm toString (1 works - 1 perfect)5/12/2013 4:30:20 PM
Test 12 copy constructor (5 pts)5/12/2013 4:30:20 PM
```