Tiffany Hernandez-tmhernan
SER 316
Assignment 2

Task 1

•Which algorithm adheres to the specification and which do not?

Special Note: there are a couple of errors that I found where the class I know adheres to the algorithm does not actually work. 1. Case of the zero points. This does not equate to an 'F' as I was told by Aanchal Mahajan like it should. 2. The hashmap's behavior does not reflect what a hashmap should do. In the case of twoStdDupName, the test only turns green if the expected grade for the one student is an 'A' but it should be an 'F' according to hashmap rules. Therefore, I have changed expected results of three test cases (oneStudentZeroP, twoStdDupName, fiveStudentZeroP) to fit the errors (not the actual results) so that I could accurately determine what is wrong with the other algorithms. I couldn't really say much from these tests about the other algorithms(because their inherently flawed), it just helped me determine which algorithm was correct in the end, which was class 4.

Moving forward, algorithm number 4 adheres to the specification and the rest 0, 1, 2, 3, 5, do not.

• Explain how the algorithms do not adhere to the specification (I want to know all the errors you were able to find).

Algorithm 0: This algorithm is only producing a correct result for a student with no points. For results for all other tests, it's inaccurately adding a phantom student of a grade 'D'.

Algorithm 1: This algorithm produces different outcomes for all students with no points or with negative points. For the negative point situations, the algorithm counts them as positive and then those grades are calculated in when they should be discarded. It then crashes when students have zero points. I suspect the algorithm also does not include the boundary zero for its points, therefore I suspect the point boundary is off here.

Algorithm 2: Considers all negatives, A grades (fiveStudentAllNeg), if there are nothing but negative grades but if you test a mix of positive and negative numbers, it counts the negative numbers as positive and correctly grades them (fiveStudentNegPos). If you have one grade in as an A, it will always count it as a failing grade. This algorithm was the most challenging.

Algorithm 3: This algorithm is written correctly with the exception of the no points test. It throws a 'null' instead of a Null Pointer Exception like the specification says it should.

Algorithm 5: For all positive point values, it adds on a grade 'Z' value with 3 students which shouldn't exist according to the specifications. Otherwise, it calculates the negative values correctly.

• Explain which test cases you chose and why.
All of my tests are in my pre-written notes below and the methods are in parenthesis. I will list there why I chose them in purple color.

Notes Before Start of Assignment:

<u>Edge cases</u>
- One student with zero points (oneStudentZeroP)
  I chose this because it was an edge case and I needed to make sure the algorithm didn't do anything funny with just one student with an zero 'F' grade which was a boundary.
- Five student with zero points (fiveStudentZeroP)

I chose this for the same reason as above but I needed to make sure it treated multiple students the same as above. Tweeked the other variable (num of students) to make sure the rule of grading at the boundary zero worked.

- five students – edge of each grade (fiveStudentEdge)

The edge case is mandatory to make sure boundaries were correct. I tested on one side of the boundaries, where each letter grade cuts off and I tested the other side (opposite).

-five students on opposite edge of each grade (fiveStudentEdgeOpp)

The edge case is mandatory to make sure boundaries were correct. I tested on one side of the boundaries, where each letter grade cuts off and I tested the other side (opposite).

-Decimal points – not possible to test

Special Cases- (discarded negatives)

-Negative numbers/Pos numbers mixed (fiveStudentNegPos)

I wanted to test to see if the behavior changed if a negative was mixed with a positive number as oppose to all negatives and all positives. And it did in class 2 although that one was most challenging.

- Five student with all neg points (fiveStudentAllNeg)

I chose this because it was an special case and I needed to make sure the algorithm discarded it, leaving it with no grades.

- One student with a negative score (negOneStudent)

I chose this for the same reason as above but I needed to make sure it treated multiple students the same as above. Tweeked the other variable to make sure the rule of discarding all negatives worked.

-Class 2 - This class was a special case for class 2 to learn more about what it did with single student A grades.

-Class 2 F - This class was a special case for class 2 to learn more about what it did with single student A grades.

Typical Values

-One student (oneStudent)

Provided

-Two students (twoStudent)

Provided

-Three students (threeStudent)

I chose this because it should be tested as a typical value as students should be slowly increased to catch any errors that have to deal with the number of students.

-Four students (fourStudent)

I chose this because it should be tested as a typical value as students should be slowly increased to catch any errors that have to deal with the number of students.

-Five Students (fiveStudent)

I chose this because it should be tested as a typical value as students should be slowly increased to catch any errors that have to deal with the number of students.

-Five Students (fiveStudentScale)

-grade values on non-typical scale (not out of 100 points)

I chose this because the point scale should work off of any high positive number so to deviate from the standard 100 points structure is good just in case there are any errors in evaluating the point scale.

Invalid/Unusual Input Values

-No class- not possible to test

-No points/no name, set_points is not called (noPoints)

To make sure the Null Pointer Exception was called correctly

-Number's for name- not possible to test

-Chars for numbers- not possible to test

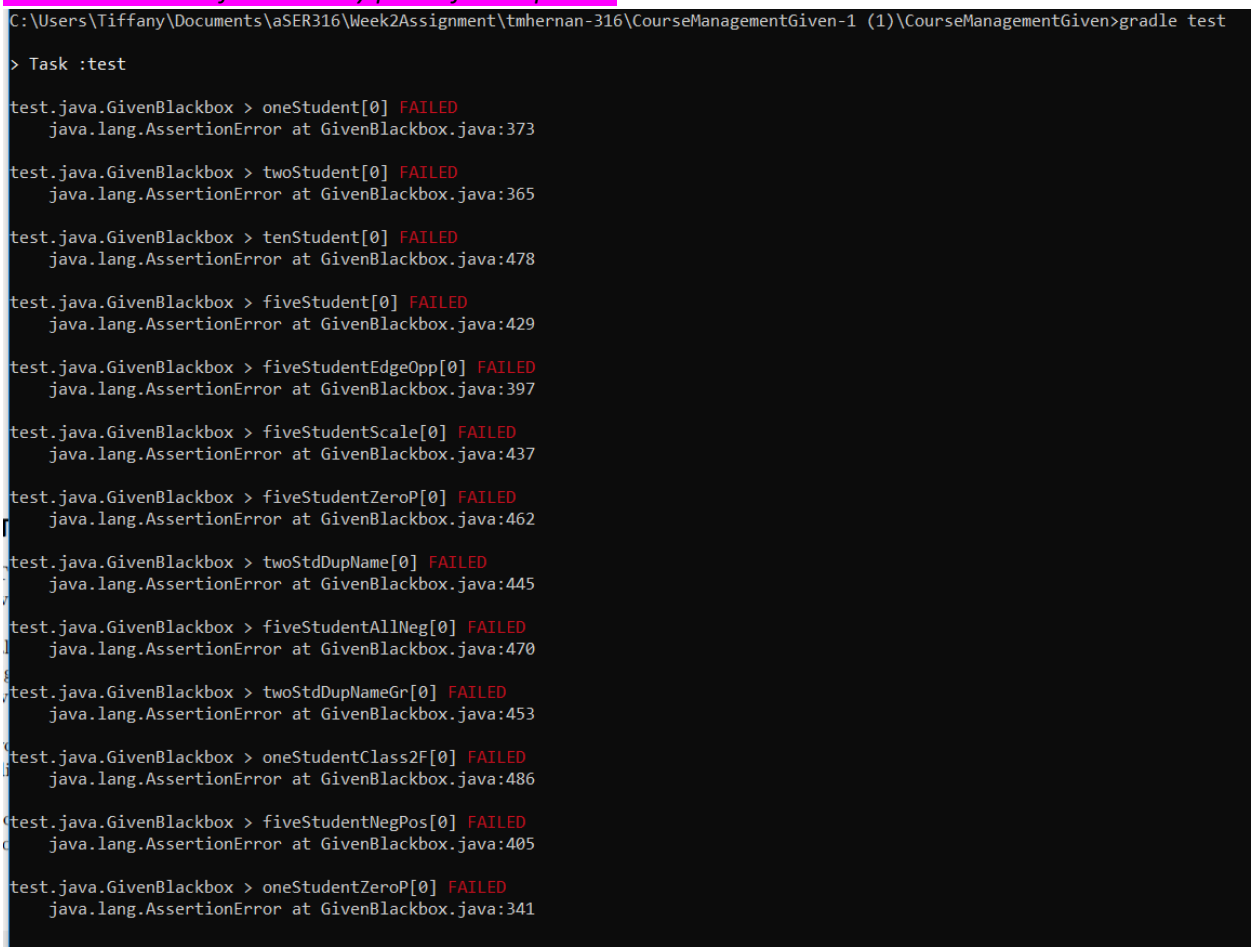
-Duplicate name, different scores (twoStdDupName)

To make sure the hashtable works which it doesn't.

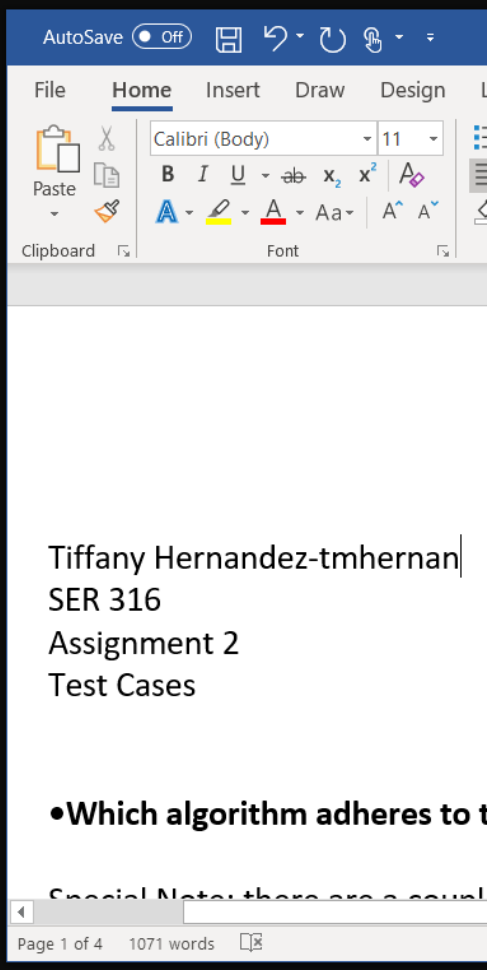
-Duplicate names and grades (twoStdDupNameGr)

To make sure the hastable worked in a way where it discarded/replaced a duplicate

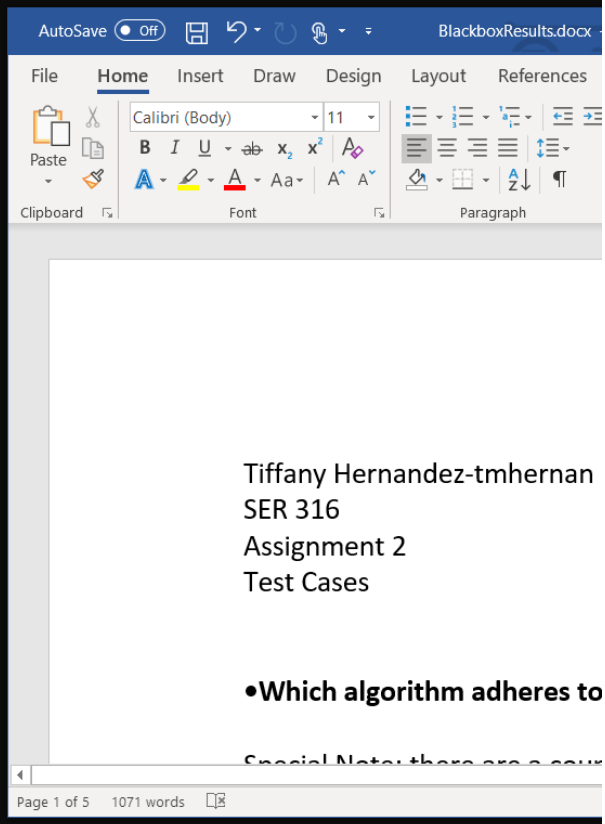
ASU user-name in file directory path of most photos



```
test.java.GivenBlackbox > threeStudent[0] FAILED
    java.lang.AssertionError at GivenBlackbox.java:413

test.java.GivenBlackbox > fourStudent[0] FAILED
    java.lang.AssertionError at GivenBlackbox.java:421

test.java.GivenBlackbox > fiveStudentEdge[0] FAILED
    java.lang.AssertionError at GivenBlackbox.java:389

test.java.GivenBlackbox > oneStudentClass2[0] FAILED
    java.lang.AssertionError at GivenBlackbox.java:381

test.java.GivenBlackbox > negOneStudent[0] FAILED
    java.lang.AssertionError at GivenBlackbox.java:349

test.java.GivenBlackbox > fiveStudentZeroP[1] FAILED
    java.lang.ArithmeticException at GivenBlackbox.java:459

test.java.GivenBlackbox > fiveStudentAllNeg[1] FAILED
    java.lang.AssertionError at GivenBlackbox.java:470

test.java.GivenBlackbox > fiveStudentNegPos[1] FAILED
    java.lang.AssertionError at GivenBlackbox.java:405

test.java.GivenBlackbox > noPoints[1] FAILED
    java.lang.AssertionError

test.java.GivenBlackbox > oneStudentZeroP[1] FAILED
    java.lang.ArithmeticException at GivenBlackbox.java:338

test.java.GivenBlackbox > negOneStudent[1] FAILED
    java.lang.AssertionError at GivenBlackbox.java:349

test.java.GivenBlackbox > oneStudent[2] FAILED
    java.lang.AssertionError at GivenBlackbox.java:373

test.java.GivenBlackbox > fiveStudentZeroP[2] FAILED
    java.lang.ArithmeticException at GivenBlackbox.java:459

test.java.GivenBlackbox > twoStdDupName[2] FAILED
    java.lang.AssertionError at GivenBlackbox.java:445

test.java.GivenBlackbox > fiveStudentAllNeg[2] FAILED
    java.lang.AssertionError at GivenBlackbox.java:470

test.java.GivenBlackbox > twoStdDupNameGr[2] FAILED
    java.lang.AssertionError at GivenBlackbox.java:453
```



```
test.java.GivenBlackbox > fiveStudentAllNeg[2] FAILED
    java.lang.AssertionError at GivenBlackbox.java:470

test.java.GivenBlackbox > twoStdDupNameGr[2] FAILED
    java.lang.AssertionError at GivenBlackbox.java:453

test.java.GivenBlackbox > oneStudentClass2F[2] FAILED
    java.lang.AssertionError at GivenBlackbox.java:486

test.java.GivenBlackbox > fiveStudentNegPos[2] FAILED
    java.lang.AssertionError at GivenBlackbox.java:405

test.java.GivenBlackbox > oneStudentZeroP[2] FAILED
    java.lang.AssertionError at GivenBlackbox.java:341

test.java.GivenBlackbox > oneStudentClass2[2] FAILED
    java.lang.AssertionError at GivenBlackbox.java:381

test.java.GivenBlackbox > negOneStudent[2] FAILED
    java.lang.AssertionError at GivenBlackbox.java:349

test.java.GivenBlackbox > noPoints[3] FAILED
    java.lang.AssertionError

test.java.GivenBlackbox > oneStudent[5] FAILED
    java.lang.AssertionError at GivenBlackbox.java:373

test.java.GivenBlackbox > twoStudent[5] FAILED
    java.lang.AssertionError at GivenBlackbox.java:365

test.java.GivenBlackbox > tenStudent[5] FAILED
    java.lang.AssertionError at GivenBlackbox.java:478

test.java.GivenBlackbox > fiveStudent[5] FAILED
    java.lang.AssertionError at GivenBlackbox.java:429

test.java.GivenBlackbox > fiveStudentEdgeOpp[5] FAILED
    java.lang.AssertionError at GivenBlackbox.java:397

test.java.GivenBlackbox > fiveStudentScale[5] FAILED
    java.lang.AssertionError at GivenBlackbox.java:437

test.java.GivenBlackbox > twoStdDupName[5] FAILED
    java.lang.AssertionError at GivenBlackbox.java:445

test.java.GivenBlackbox > twoStdDupNameGr[5] FAILED
    java.lang.AssertionError at GivenBlackbox.java:453
```




```
    java.lang.AssertionError at GivenBlackbox.java:453

test.java.GivenBlackbox > oneStudentClass2F[5] FAILED
    java.lang.AssertionError at GivenBlackbox.java:486

test.java.GivenBlackbox > fiveStudentNegPos[5] FAILED
    java.lang.AssertionError at GivenBlackbox.java:405

test.java.GivenBlackbox > threeStudent[5] FAILED
    java.lang.AssertionError at GivenBlackbox.java:413

test.java.GivenBlackbox > fourStudent[5] FAILED
    java.lang.AssertionError at GivenBlackbox.java:421

test.java.GivenBlackbox > fiveStudentEdge[5] FAILED
    java.lang.AssertionError at GivenBlackbox.java:389

test.java.GivenBlackbox > oneStudentClass2[5] FAILED
    java.lang.AssertionError at GivenBlackbox.java:381

115 tests completed, 49 failed

> Task :test FAILED

FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':test'.
> There were failing tests. See the report at: file:///C:/Users/Tiffany/Documents/aSER316/Week2Assignment/tmhernan-316/CourseManagementGiven-1%20(1)/CourseManagementGiven/build/reports/tests/test/index.html

* Try:
Run with --stacktrace option to get the stack trace. Run with --info or --debug option to get more log output. Run with --scan to get full insights.

* Get more help at https://help.gradle.org

BUILD FAILED in 6s
3 actionable tasks: 2 executed, 1 up-to-date

C:\Users\Tiffany\Documents\aSER316\Week2Assignment\tmhernan-316\CourseManagementGiven-1 (1)\CourseManagementGiven>
```

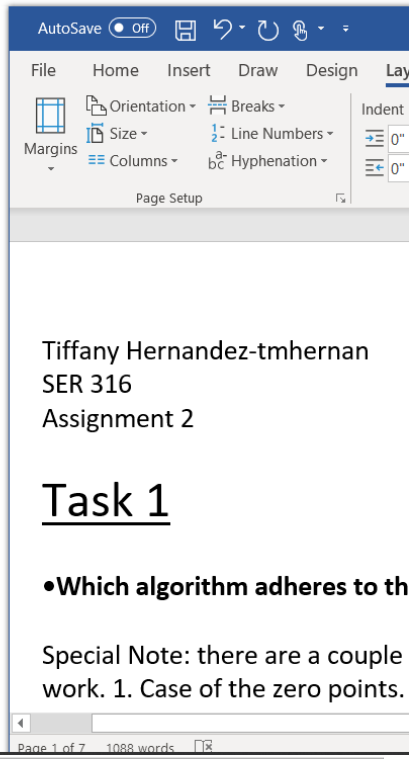
Test Summary

115	49	0	0.194s
tests	failures	ignored	duration

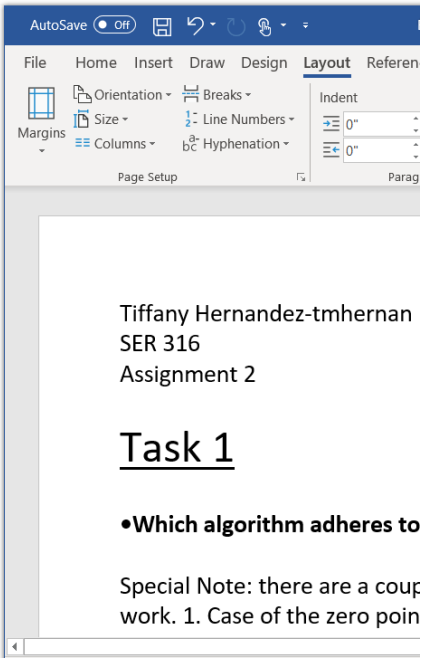
57% successful

Failed tests | Packages | Classes

GivenBlackbox. fiveStudentAllNeg[0]
GivenBlackbox. fiveStudentAllNeg[1]
GivenBlackbox. fiveStudentAllNeg[2]
GivenBlackbox. fiveStudentEdgeOpp[0]
GivenBlackbox. fiveStudentEdgeOpp[5]
GivenBlackbox. fiveStudentEdge[0]
GivenBlackbox. fiveStudentEdge[5]
GivenBlackbox. fiveStudentNegPos[0]
GivenBlackbox. fiveStudentNegPos[1]
GivenBlackbox. fiveStudentNegPos[2]
GivenBlackbox. fiveStudentNegPos[5]
GivenBlackbox. fiveStudentScale[0]
GivenBlackbox. fiveStudentScale[5]
GivenBlackbox. fiveStudentZeroP[0]
GivenBlackbox. fiveStudentZeroP[1]
GivenBlackbox. fiveStudentZeroP[2]
GivenBlackbox. fiveStudent[0]
GivenBlackbox. fiveStudent[5]
GivenBlackbox. fourStudent[0]
GivenBlackbox. fourStudent[5]
GivenBlackbox. negOneStudent[0]
GivenBlackbox. negOneStudent[1]
GivenBlackbox. negOneStudent[2]
GivenBlackbox. noPoints[1]
GivenBlackbox. noPoints[3]
GivenBlackbox. oneStudentClass2F[0]
GivenBlackbox. oneStudentClass2F[2]
GivenBlackbox. oneStudentClass2F[5]
GivenBlackbox. oneStudentClass2[0]
GivenBlackbox. oneStudentClass2[2]



GivenBlackbox. fiveStudentScale[5]
GivenBlackbox. fiveStudentZeroP[0]
GivenBlackbox. fiveStudentZeroP[1]
GivenBlackbox. fiveStudentZeroP[2]
GivenBlackbox. fiveStudent[0]
GivenBlackbox. fiveStudent[5]
GivenBlackbox. fourStudent[0]
GivenBlackbox. fourStudent[5]
GivenBlackbox. negOneStudent[0]
GivenBlackbox. negOneStudent[1]
GivenBlackbox. negOneStudent[2]
GivenBlackbox. noPoints[1]
GivenBlackbox. noPoints[3]
GivenBlackbox. oneStudentClass2F[0]
GivenBlackbox. oneStudentClass2F[2]
GivenBlackbox. oneStudentClass2F[5]
GivenBlackbox. oneStudentClass2[0]
GivenBlackbox. oneStudentClass2[2]
GivenBlackbox. oneStudentClass2[5]
GivenBlackbox. oneStudentZeroP[0]
GivenBlackbox. oneStudentZeroP[1]
GivenBlackbox. oneStudentZeroP[2]
GivenBlackbox. oneStudent[0]
GivenBlackbox. oneStudent[2]
GivenBlackbox. oneStudent[5]
GivenBlackbox. tenStudent[0]
GivenBlackbox. tenStudent[5]
GivenBlackbox. threeStudent[0]
GivenBlackbox. threeStudent[5]
GivenBlackbox. twoStdDupNameGr[0]
GivenBlackbox. twoStdDupNameGr[2]
GivenBlackbox. twoStdDupNameGr[5]
GivenBlackbox. twoStdDupName[0]
GivenBlackbox. twoStdDupName[2]
GivenBlackbox. twoStdDupName[5]
GivenBlackbox. twoStudent[0]
GivenBlackbox. twoStudent[5]



Generated by Gradle 5.2.1 at Mar 26, 2019, 9:31:56 AM

Task 2

ASU user-name in file directory path

```
C:\Users\Tiffany\Documents\aSER316\Week2Assignment\tmhernan-316\CourseManagementGiven-1 (1)\CourseManagementGiven>gradle jacocoTestReport

BUILD SUCCESSFUL in 4s
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\Tiffany\Documents\aSER316\Week2Assignment\tmhernan-316\CourseManagementGiven-1 (1)\CourseManagementGiven>
```

CourseManagementGiven Sessions

CourseManagementGiven

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
main.java		12%		8%	26	30	70	86	20	24	3	4
Total	378 of 433	12%	11 of 12	8%	26	30	70	86	20	24	3	4

Created with JaCoCo 0.8.2.201808211720