Brendan Jang
CS 362
Assignment 4

Documentation

Unit Testing

In this assignment, I wrote Unit Tests to identify and correct the bugs in classroom_manager.py – specifically the two classes included in this file: Student and Assignment.

Student class:

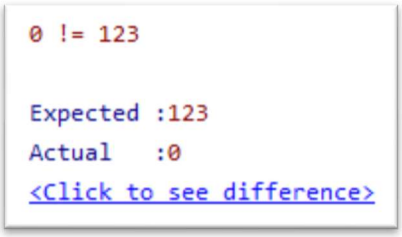
- In test_student_constructor(), the code constructs a Student object. It then checks that the id, first_name, last_name, and assignments instance variables are set correctly based on the constructor parameters.
- In test_student_get_full_name(), the code constructs a Student object. It then calls get_full_name() to check that the String returned matches the expected result.
- In test_student_submit_assignment(), the code constructs a Student object and an Assignment object. It then calls the submit_assignment() method and checks that the assignments instance variable is updated correctly.
- In test_student_get_assignments(), the code constructs a Student object and two Assignment objects. It calls the submit_assignment() method with both assignments and then calls get_assignments() to check that the assignments are returned properly.
- In test_student_get_assignment_name(), the code constructs a Student object and two Assignment objects. It calls the submit_assignment() method with both assignments and then calls get_assignment(name) to verify that the correct assignment is returned. It also calls get_assignment(name) on a nonexistent assignment to verify that None is returned.
- In test_student_get_average(), the code constructs a Student object and two Assignment objects, assigning grades to each of them. It calls the submit_assignment() method with both assignments and then calls get_average() to verify that the correct average is returned. It then constructs a third assignment with a None grade and calls submit_assignment() again to verify that the average does not change.
- In test_student_remove_assignment(), the code constructs a Student object and two Assignment objects. It calls remove_assignment() on both assignment objects and looks at the assignments instance variable to verify that the assignments were removed correctly. The test also tries to remove a nonexistent assignment to verify functionality.

Assignment class:

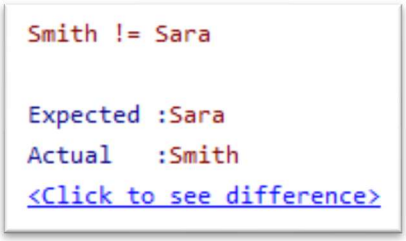
- In test_assignment_constructor(), the code constructs an Assignment object. It then checks that the name, max_score, and grade instance variables are set correctly based on the constructor parameters.
- In test_assignment_assign_grade(), the code constructs an Assignment object. It then calls assign_grade() to assign a grade to the assignment which is below the max score. It checks that the grade is set correctly. It again calls assign_grade() to assign a grade to the assignment which is above the max score. It checks that the grade is updated to None.

Debugging

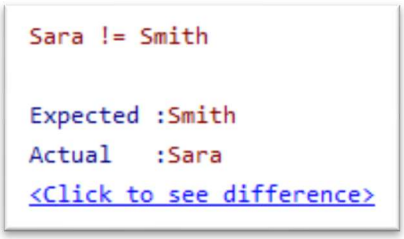
- In test_student_constructor(), I found errors in the constructor of the Student class.
    - On line 7, self.id is defaulting to 0 rather than being set to id.



    - On line 8, self.first_name is being set to last_name rather than being set to first_name.



    - On line 9, self.last_name is being set to first_name rather than being set to last_name.



    - On line 10, self.assignmentss is being initialized instead of self.assignments.

```
  File "C:\Users\Brendan\PycharmProjects\CS362-W2020\projects\jangb\Assignment4\test_classroom_manager.py", line 13, in
 test_student_constructor
    self.assertEqual(student.assignments, [])
AttributeError: 'Student' object has no attribute 'assignments'
```

- In test_student_get_full_name(), I found an error in the String format being returned by the method.
    - On line 13, the code is incorrectly constructing the name as first_name,last_name instead of first_name + " " + last_name.

```
Sara,Smith != Sara Smith

Expected :Sara Smith
Actual   :Sara,Smith
<Click to see difference>
```

- In test_student_submit_assignment(), I found an error in how assignments are being added to the Student object.
    - On line 17, the code is incorrectly adding the assignment to the assignments instance variable a second time.

```
[<classroom_manager.Assignment object at 0x055E5850>,
 <classroom_manager.Assignment object at 0x055E5850>] != [<classroom_manager.Assignment object at 0x055E5850>]

<Click to see difference>
```

- In test_student_get_assignments(), I found an error in how assignments are being returned by the Student object.
    - On line 20, the code is incorrectly only the first assignment in the list using slicing, which is not needed.

```
[<classroom_manager.Assignment object at 0x05835930>] != [<classroom_manager.Assignment object at 0x05835930>,
 <classroom_manager.Assignment object at 0x058356B0>]

<Click to see difference>
```

- In test_student_get_assignment_name(), I found an error in how the assignment is identified to be returned.
    - On line 24, the code is incorrectly checking that the name of the assignment is 'name' rather than correctly comparing it to the value of the variable name.

```
None != <classroom_manager.Assignment object at 0x051C4970>

Expected :<classroom_manager.Assignment object at 0x051C4970>
Actual   :None
<Click to see difference>
```

- In test_student_get_average(), I found a few errors in how the average is calculated and then returned.
    - On line 29, the code incorrectly declares total_assignments as total_assignmentss.

```
27      def get_average(self):
28          sum_grades = 0
29          total_assignmentss = 0
30          for a in self.assignments:
31              if a.grade != None:
32                  sum_grades = sum_grades - a.grade
33                  total_assignments = total_assignments + 11
```

    - On line 32, the code incorrectly updates sum_grades by subtracting the assignment grade, whereas it should add the assignment grade.

```
32                  sum_grades = sum_grades - a.grade
```

    - On line 33, the code incorrectly increases total_assignments by 11 for each assignment, instead of the 1 it should increase by.

```
33                  total_assignments = total_assignments + 11
```

    - On line 34, average is incorrectly calculated as total_assignments / sum_grades, whereas it should be sum_grades / total_assignments.

```
11.0 != 70

Expected :70
Actual   :11.0
<Click to see difference>
```

- In test_student_remove_assignment(), I found a few errors in how assignments are removed.
    - On line 39, the code is incorrectly checking that the name of the assignment is 'name' rather than correctly comparing it to the value of the variable name.

```
37      def remove_assignment(self, name):
38          for a in self.assignments:
39              if a.name == 'name':
40                  del name
```

    - On line 40, the code is using the wrong operator to remove the entry from the list. Rather than del, it should use list.remove(entry). Here, del name should be replaced by self.assignments.remove(a).

```
[<classroom_manager.Assignment object at 0x05DA5990>,
 <classroom_manager.Assignment object at 0x05DA5690>] != [<classroom_manager.Assignment object at 0x05DA5690>]

<Click to see difference>
```

- In test_assignment_constructor(), I found an error in the constructor of the Student class.
    - On line 47, the grade instance variable is being initialized incorrectly to -1, but it should be initialized to None.

```
-1 != None

Expected :None
Actual   :-1
<Click to see difference>
```

- In test_assignment_assign_grade(), I found several errors in how the grade is assigned to the Assignment object.
    - On line 50, the code mistakenly uses the "==" operator for assignment, instead of the "=" operator.

```
50              self.grade == grade
```

    - On line 51, the code incorrectly uses the ">=" comparator for checking grades over max_score, instead of the ">" comparator.

```
51 ●          if grade >= self.max_score:
```

- On line 52, the code wrongly sets grade to -1 instead of setting self.grade to None in the case where a grade is greater than max_score.

```
69 != None

Expected :None
Actual   :69
<Click to see difference>
```

Code Coverage

Running Python code coverage demonstrates that the tests achieve 100% code coverage:

```
(venv) C:\Users\Brendan\PycharmProjects\CS362-W2020\projects\jangb\Assignment4>coverage run test_classroom_manager.py
.........
----------------------------------------------------------------------
Ran 9 tests in 0.001s

OK

(venv) C:\Users\Brendan\PycharmProjects\CS362-W2020\projects\jangb\Assignment4>coverage report
Name                       Stmts   Miss  Cover
----------------------------------------------
classroom_manager.py          38      0   100%
test_classroom_manager.py     73      0   100%
----------------------------------------------
TOTAL                        111      0   100%
```

```
C:\Users\Brendan\PycharmProjects\CS362-W2020\projects\jangb\Assignment4>git status
On branch jangb-assignment-4
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   ../../../.idea/CS362-W2020.iml
        modified:   ../../../.idea/misc.xml

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        ./

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\Brendan\PycharmProjects\CS362-W2020\projects\jangb\Assignment4>dir
 Volume in drive C has no label.
 Volume Serial Number is 4847-3C13

 Directory of C:\Users\Brendan\PycharmProjects\CS362-W2020\projects\jangb\Assignment4

02/23/2020  06:45 PM    <DIR>          .
02/23/2020  06:45 PM    <DIR>          ..
02/23/2020  06:08 PM            53,248 .coverage
02/23/2020  06:45 PM             1,439 classroom_manager.py
02/23/2020  06:45 PM             4,138 test_classroom_manager.py
02/23/2020  06:07 PM    <DIR>          __pycache__
               3 File(s)         58,825 bytes
               3 Dir(s)  94,714,916,864 bytes free

C:\Users\Brendan\PycharmProjects\CS362-W2020\projects\jangb\Assignment4>
```