**StudentCollectionTest Corrections**

**Shawn Aparo**

1. **testAddStudent(), Line 435;** Attempting to add a Student object to a full collection. Placed within a try-catch block as StudentCollectionException is expected to be thrown.
2. **testRetrieveStudentBySID(), Line 499;** Added a scenario to test an empty string. We did not include this in our first submission.
3. **testRetrieveStudentBySID(), Line 503;** Changed the invalid SID string from what we included in our first submission.
4. **testRetrieveStudentBySID(), Line 509;** Added a test scenario to test passing a null value. Placed in a try-catch block as an exception is expected.
5. **testRemoveStudentBySID();** I reimplemented this entire test case as I felt our previous implementation has some unnecessary additional test scenarios. I felt that it made sense for me to reimplement this in my own style, as I had a difficult time following my partner’s logic. This is in no way a criticism of my partner’s work, style, or logic. I simply have a different thought process and felt it was important for my own understanding to implement this myself.
6. **testCreateIterator():** I re-organized this test case to improve the readability.
7. **testCreateIterator(), Line 768;** I added a test scenario to confirm that the remove operation is not supported.
8. **testGetSpacesRemaining();** I re-organized this test case to improve the readability.
9. **testRetrieveCollectionFromDisk(), Line 1170;** Added a test scenario to confirm that an object of type student, but with a different SVUID, will throw an exception.
10. **testRetrieveCollectionFromDisk(), Line 1267;** The initial implementation of this test scenario compared the retrieved student object to the current student object in the reference collection using assertEquals. Given that the memory locations are not the same, the assertion fails. I’m not sure why my partner didn’t adjust this after he implemented, however, I changed it to compare the values of each Student object’s fields.