Kris Williams

Kxw62

EE360C

**Program 3 Testing Report**

To test my program for correctness, I decided to implement a simple, static, three-by-four array to test various dynamic allocations of hours to the final solution. By using a relatively small amount of hours available to allocate, it would be straightforward to verify that my program is producing the correct resultant array by hand. Additionally, I was able to test various grade saturation points, as well as easily testing corner cases like classes which never improve despite increasing study hours. By using a static array, I was additionally able to call on my friends to help me come up with additional test arrays.

I found the small array to be effective for testing these conditions quickly and easily, without the burden of creating several additional classes. Admittedly, the major shortfall of my testing approach is that I largely neglected larger test cases and unusual, formulaic approaches to the grade function. Ultimately, I thought that testing a thorough variety of different situations on a small scale would translate well to a larger test scenario.

Being able to test a variety of situations relatively quickly benefited me, since I was able to find relatively quickly that I was not correctly tracking how to allocate hours to the proper projects correctly. Due to a programmatic error in indexing an array, I saw that I would only log the proper hours for the final project which required hours. Essentially, any scenario which required hours to be allocated to more than one project would produce an incorrect output from my code. Although this may seem like a major error to overlook, without thorough testing, I may not have found this bug.

In conclusion, I valued and prioritized the ability to “shotgun” test, and wanted to test as many different scenarios as quickly as possible. While my submitted test case only has one such array that I used to test, I valued the ability to collaborate with my colleagues to produce more arrays.