**Deliverable 2 – CitySim9004**

**CS1632 with Bill Laboon**

Github URL:

https://github.com/JulianMonticelli/Deliverable\_2\_Unit\_Tests

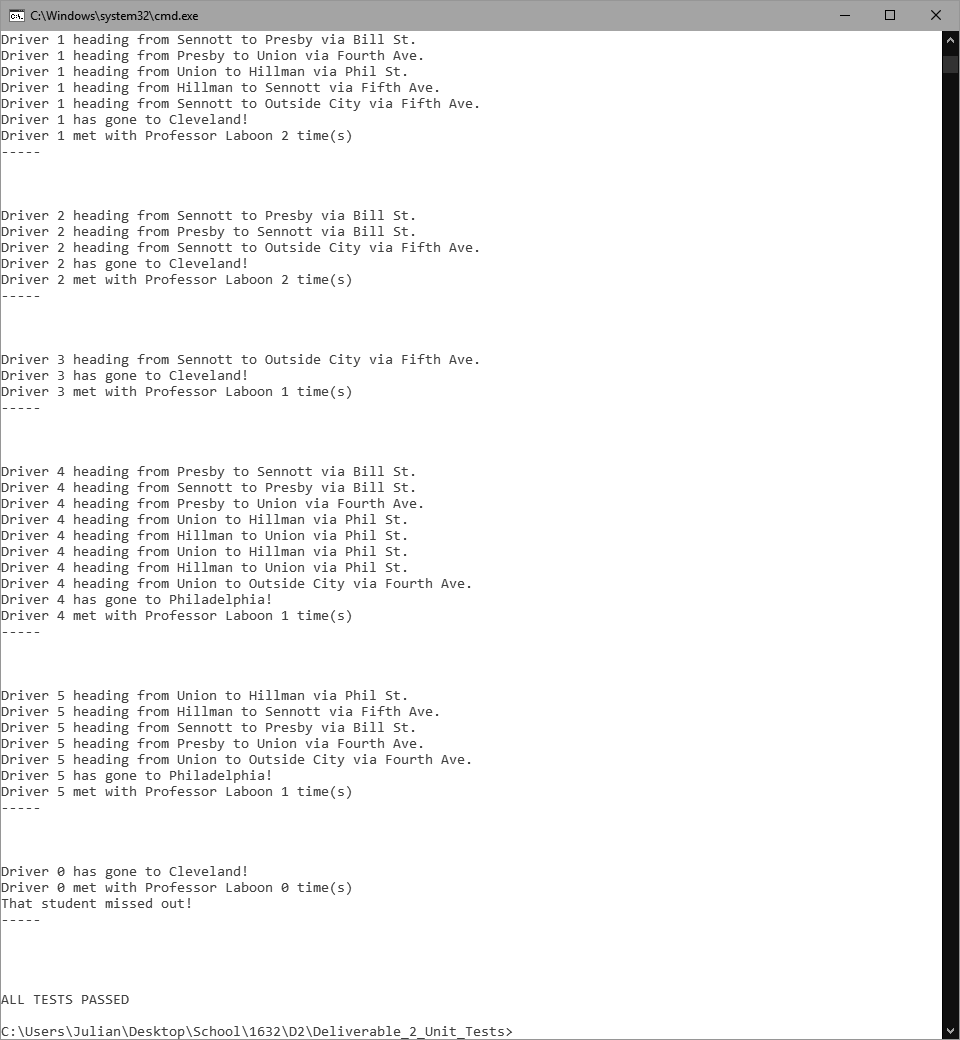
Julian Monticelli

During my testing and development of CitySim9004, I noticed that development of a test plan is actually quite difficult. It is not so much that designing individual tests is so hard, but more that designing a suite of tests while coordinating development to allow said tests to work is challenging.

Ultimately, a good deal of my tests were intuitive – use object oriented programming ideals for Drivers, and unit test each traversal from street to street to guarantee they work as intended, check to see what output is given under a certain circumstance, et cetera. It was when I ran out of those things to test that testing seemed much more daunting than before.

Although using a RNG test double was easy, and done in numerous tests, I struggled to find where to stub methods. In fact, even though I wrote a large portion of my tests while writing code, I still was unsure where to put the recommended three method stubs. I finally decided to add them to (I think) ten tests to satisfy a concern that failure cases would fail. Also, I added them to the tests where I was in need of boundary values, instead of setting boundary values.

All in all, I don’t know how much more I could have designed this program for testing. I feel as though I did a pretty good job for my first program that I designed specifically to be tested, though I’m sure that there are plenty of places I could have added new tests. I have a lot of respect for anyone who can make several hundred tests for their program, because each test took me longer than I would have imagined it to make, although, that may just be because I refactored some code here and there to allow the tests to perform as expected and pass results as intended.



(My test runs the program itself to test how many drivers there are and in what order they occur. This is as the largest screenshot I can get without going over my screen length.)