Michael Oles

**CS 1632 – DELIVERABLE 2: Unit Testing CitySim9003**

<https://github.com/mikeoles/CitySim9003>

Description

When creating the actual program, I struggled to figure out how to best create this in with an object oriented design. I first wanted to create locations that held a driver inside of them and passed the driver between locations but since I needed some of the data from the individual driver, I found that it worked better when I focused the program around the driver object and gave it a location field. I also struggled because I needed not only to connect the locations to each other, but also to know which streets connected them. This is the reason I added the path object: so I would be able to know which streets were being used to traverse between locations.

I realized that some of my methods would need to be changed to work well for tests. For example, my checkArgs() method original called System.exit(0) when there was an invalid argument. This was difficult to test because it would not return a value. Instead, whenever there was an invalid argument found, I would throw and exception and then I could check that an exception was thrown by the method in my test and deal with it in the main program.

I definitely had trouble figuring out when to use mocks. I think it was because this is a relatively small program and I felt confident my code would work right so I didn’t see the need for them. For example, when I first tested the coffee function, I just created a new location with (“Coffee”) in the constructor. Then I realized that me doing that is assuming that the location’s getName() function will work right. Instead I stubbed the getName() function to return the string I needed for the test. Since it was something as simple as a getter it was hard for me to realize when I needed to stub a method and not rely on it working.

Another major mistake I made was writing too much code before writing mocks. My outside city class manually checked if the city name was equal to “Philadelphia” or “Cleveland” and then returned that it was outside of the city but I realized this was probably not good object oriented code and also not very easy to test. So I instead passed in a list of outside cities which I could stub to return a certain city which would be either be the same as the city passed in or different to check both the true and false cases.

Although I was able to solve those problems, I had trouble figuring out a good test for one method: the getRandomPath(int seed) method. I think this is because this method relies on fields that have to be set after the program starts because they depend on the specific orientation of the map. I probably need a way to abstract this out of the class but I could not figure out any reasonable way to do that so instead I tested that when there is a null Path that it throws an illegalStateException().

Executed Unit Tests

