Samuel Bailey

CS-230

Project Two

**Summary**

Project One was created directly from the rubric requirements. Every line of code and every feature was based on the rubric. Each module piece had a specific limit for how long the data can be and the @Test checks the limits of the objects.

There is multiple ways to create JUnits, The easiest way is to just use the standard library that they give to you. Using assert functions made creating the code very easy. The hardest part was reading through the documentation to understand the library.

@Test  
public void validID(){  
 Date date2 = new Date();  
 Appointment appointment2 = new Appointment("00000001", date2, "example");  
 *assertNotNull*(appointment2.getId());  
 *assertTrue*(exampleId.length() >= appointment2.getId().length());  
}

In the Test function above we can walk through it and see there is an object that uses assertNotNull which checks if the object is null. This test case is only checking the ID. Then assertTrue checks if the following statement is true. For context the exampleId is a String with 10 characters. This is the limit for how long the iD can be. So in the assertTrue if the appointment.getID().length() is longer than 10 characters then it will fail the test. It’s a different way than an if statement but that’s what this class was about.

**Reflection**

For techniques I wanted the test cases to be simple and short. There is other ways to use test cases like with fail() or success(). I could have made some long drawn if statement checking every condition and then putting fail or success at the end for it to pass the case. Although this works it’s not common. Using the library given is much more handy and efficient once you learn what it can do.

Other techniques are manual testing, this is where you make a print statement after each step to walk through the code to make sure there is no errors. This just takes far to much time. Any company larger than 20 people will more than likely have software testers making this method unused.

**Mindset**

It was important that these test actually did their job. When making testing it is import to truly understand the logic that is going on behind the scenes to make these test properly work.

When I make my own code I know what It can do so when building tests I’ll probably make some sort of test code that will almost always pass. I can almost always say that I think it would be best for someone else to test the code. When creating something that is going to go to the public or an internal company for people to use, we must prepare for the worst. Imagine everyone is trying to reverse engineer the project to get access to the data inside. This is something to be very aware of when making code. That is why not everyone publicly releases their code. Through testing and privacy is how good applications are made.