**Sequence Diagrams**

Sequence diagrams are a method of documenting and visualizing the handling of messages and calling processes within an event sequence. Below is a link I used for reading more about sequence diagrams.

<https://en.wikipedia.org/wiki/Sequence_diagram>

I've attached a set of sequence diagrams I created for CIT-262 last semester which helped me refresh my memory regarding this subject. The file is [Group 4 - Lab 5.pdf](https://github.com/hodges-olan/CIT360-Portfolio/blob/master/CIT360-Portfolio/src/SequenceDiagram/Group%204%20-%20Lab%205.pdf)

**MVC (Model, View, Control)**

Brandon Urednick presented over MVC this week and explained the purpose behind MVC. He used the link at http://www.w3schools.com/aspnet/mvc\_intro.asp to show us the meaning behind each of the layers. I can see how this is a popular method, as it allows you to break down the basic functionality of most programs to this separate sections. This was the same way we split up the game that we created in CIT-260, so it was somewhat familiar to most of us. I'm working to get the code the Brandon Urednick had so that I can post it to this note.

[MVC Code – Brandon Urednick Example](https://github.com/hodges-olan/CIT360-Portfolio/blob/master/CIT360-Portfolio/src/ModelViewControllerPattern/MVC.java)

**JUnit Tests**

This week was the first week of assignments and I decided to study JUnit Tests more in depth. From the previous CIT 260 class, the JUnit Tests were made so that you could not only verify that your metholodgy was correct, but also that your algorithms were processing correctly. We would provide a type of truth table for the algorithm so that we could test both the limits of the algorithm, as well as things outside of the expected boundaries to verify that we were passing errors back appropriately to the calling code.

I found the following link which is JUnits actual website with definitions of the idioms used in JUnit tests.

<http://junit.org/junit4/>

The assertions page was the most informative this time, as I was able to see all of the verifications you could imploy with these tests. Not only could you check to make sure they were the same, but you could also check for null, not null, true, false, contains, etc.

Below is the link to the meeting where I taught regarding JUnit tests. It starts at 0:33 (the video takes you directly to the second, so you don’t have to scroll through the video)

<https://youtu.be/TP41oyVhwxY?t=33s>

Below is my testing code from the presentation.

[JUnitTests.java](https://github.com/hodges-olan/CIT360-Portfolio/blob/master/CIT360-Portfolio/src/JUnitTests/JUnitTests.java)