Mutation Rate:

1. The original mutation rate was at 53% coverage on my personal tests and 65% coverage on the original Evo suite test. By the end of mutation testing I ended up with a mutation rate of 71% for personal and when that was then combined with Evo suite testing it concluded at 91%.

2. The mutation rate was generally poor on TimeTable for both types of testing but was also the easiest to raise after figuring out where the gaps generally rested in the code. Appt had the most successful hits but was only about half covered on generated tests. By the end of testing it was CalDay that had the hardest time being bumped up in coverage.

3. Errors mostly existed within return calls, often failing to correctly align with the expected output. Though mostly resolved an interesting error that I was struggling to handle was from loop statements. It was where I spent the most time trying to correct. The loop appears to be difficult to deal with because of the requirement to check all possible conditions but may have been timing out before completion. Return statements often proved challenging due to failure in determining how to correctly reach that line of code or because I was unable to correctly predict its expected output.

Original test case:

Pit Test Coverage Report

Package Summary

edu.osu.cs362

Number of Classes	Line Coverage		Mutation Coverage	
3	94%	151/160	53%	52/99

Breakdown by Class

Name	Line Coverage		Mutation Coverage	
Appt.java	95%	58/61	63%	27/43
CalDay.java	94%	50/53	54%	14/26
TimeTable.java	93%	43/46	37%	11/30

Original Evo test case:

Pit Test Coverage Report

Package Summary

edu.osu.cs362

Number of Classes	Line Coverage		Mutation Coverage	
3	98%	157/160	65%	64/99

Breakdown by Class

Name	Line Coverage		Mutation Coverage	
Appt.java	100%	61/61	84%	36/43
CalDay.java	100%	53/53	46%	12/26
TimeTable.java	93%	43/46	53%	16/30

Final test case:

Pit Test Coverage Report

Package Summary

edu.osu.cs362

Number of Classes	Line Coverage		Mutation Coverage	
3	99%	159/160	91%	90/99

Breakdown by Class

Name	Line Coverage		Mutation Coverage	
Appt.java	100%	61/61	93%	40/43
CalDay.java	100%	53/53	81%	21/26
TimeTable.java	98%	45/46	97%	29/30

Overall Experience:

1. No issues with the mutation testing tool beyond its interactions with Evo suite. The error I had was dealing with how the auto generated code would execute which would cause PIT to crash while running. Once the separate command in Evo's header was removed the code worked as intended.

2. Mutation tool was easy to use and straightforward. I enjoyed it's explicit stating of where and what it did to help improve coverage. It was a bit frustrating that it wasn't by default capable of handling Evo but that likely is a fault of Evo suite more so than PIT. It does work weirdly in how many timeouts I received while testing. This must do with the looping components but you'd normally think there is some better method of catching the issue as opposed to just idling for a period before continuing. This also resulted in no useful information being presented regarding the loops.