Report 2
Jeff Olson

For this report, I wrote a bunch of random testing code, but I was not able to get it working to my standards before the due date. As such, I adapted my previous unit test for buycard into a more robust system that tests all of my group members following the required format.

I found several possible bugs using this new method, but it's entirely possible that this is simply due to a difference in return values for different implementations of buycard.

The following table lists coverage percentages and tests failed for each of my group members.

Dunhame	19.62%	Failed test 1
Wolfej	17.70%	Failed test 3
Randb	17.86%	Failed test 2
Wandlins	Didn't compile	Didn't compile
Olsojeff	18.83%	Passed all tests
Wesbyb	Didn't compile	Didn't compile
Parkan	21.01%	Failed test 3
Milleand	18.88%	Failed test 4
Alarkas	Didn't compile	Didn't compile

The tests were generally as follows:

- Test 1 tested to see if the function allowed you to buy a card with no coins.
- Test 2 tested to make sure you could buy a card.
- Test 3 checked that the amount of coins changed by the right amount when you bought a card
- Test 4 tested that the amount of buys you had decremented when you bought a card.
- Test 5 tested that the card was put on top of your discard pile.
- Test 6 was a test again to see if you could buy something without enough coins.

However, as previously stated, the tests rely on uniform return values for the buycard function, which is not a reality. I will be replacing all of this code with random testing in the future. Comparing game states is much less foolproof.