[illegible]
[illegible]110 59

In-class test for SOEN 345, Prof Rigby

Worth 15% of total grade. Don't cheat, 15% isn't worth the black mark on your academic record. 60 minutes. **Answer all questions in the exam booklet!**

Multiple Choice [3 marks]

Question 1: Mockito can only mock classes (True or False)

Question 2: After establishing the first and last commit as good and bad respectively, what is the **maximum** number of test executions we would need to find the culprit?

1----2----3----4----5----6----7----8----9----10----11

Question 3: Which tests should we run first?

a. The flakiest test.
b. The test that has failed the least in the past.
c. The test that has failed the most in the past.

Short Answer [4 marks]

Question 4: What are the implications of Gall's law [2 marks]

Question 5: Define flaky tests [1 mark]

Question 6: How do we solve the "Construction Blob" testing smell? [1 mark]

Coding Questions

Question 7: Write a test to characterize the following code [2 marks]

```
public float division(float numerator, float denominator) {
                if (denominator == 0) {
                        return 0;
                }
                return numerator/denominator;
        }
}
```

Question 8: Refactor, ~~Dependency Injection~~, and Mocking [6 marks]

```
public class Movie {
        ProjectorScreen s;
        Price p;
        public Movie() {
                this.s = new ProjectorScreen(16,9);
                this.p = new price();
        }
        public void playMovie(Float discount) {
                p.discount(discount);
                s.play(this);
        }
...
}
```

a. In a diagram, break the dependency on ProjectorScreen and add an iPadScreen
b. Parameterize the constructor
c. In a test, mock out the dependencies and ensure that price is discounted at 15% or .15 and that the movie plays on the screen

Question 9: ArgumentCaptor and InOrder [4 marks]

Recall this simple code written in class:

```
public void testSaleScan() {
        Display display = mock(Display.class);
        HashStorage storage = mock(HashStorage.class);
        when(storage.barcode("1A")).thenReturn("Milk, 3.99");

        Sale sale = new Sale(display, storage);
        sale.scan("1A");

        verify(display).showLine("1A");
        verify(storage).barcode("1A");
        verify(display).showLine("Milk, 3.99");
}
```

Important: you only need to re-write the lines you are changing not the entire piece of code

a. Re-write the code so that ArgumentCaptor is used in place of the item id and the order of calls are checked (ie check that storage is called before display)