Post Mortem P1
CS247 S17

Post Mortems are released after each assignment as a way of detailing the common errors that were found by the markers. By reviewing these errors, as well as the individual feedback for your assignment, you can learn which areas you can improve on for future assignments/exams.

**Code:**
- Overall, the code was written very well for this assignment.
- Many students are including sufficient comments, but there are still a few students who have no comments whatsoever.
- A few students did not create a Deck class, and instead directly held a collection of cards in their game state class. Remember to apply Single Responsibility where possible, to encapsulate and reduce coupling.

**UML:**
- Some students did not include all of their program’s classes in their UML.
- A few students drew their diagrams with their relationship arrows pointing in the wrong direction. Inheritance arrows should point to the base class, and aggregation/composition diamonds should be on the side of the “owning” class.
- Some students grouped multiple class relationships into a single association/aggregation/composition, as opposed to showing each relationship separately. An example of this would be the Player class needing both a hand of Cards, and a collection of discarded Cards. The purpose of a UML is to describe how classes interact with each other, so your diagram should explicitly reflect each of these relevant relationships.
- Some students did not specify navigabilities and multiplicities as clear as possible.
- Some students forgot to indicate methods’ visibilities (private, public or protected) in their UML.
- A number of class relationships were forgotten. Some of the more common ones were:
  - A player having a discarded pile of Cards
  - Some class that keeps track of any Cards that have been played
  - Main having a relationship to at least one of the other classes.