Jennifer Swofford  
APC 390, Assignment 2

This topic of what is aggregation and what is composition has been rather difficult for me to comprehend. I understand that both are illustrated by the has-a relationship. I also conceptually understand that composition is a tightly coupled has-a relationship in which Class B object is destroyed when Class A is object destroyed. While with aggregation, the class objects can live independently. The real struggle has been in the understanding of the implementation and how they’re actually different in the code. Some of the analogies I read (like car & engine) didn’t help because I found myself making philosophical arguments as to whether the car and engine could exist independently. So after rewatching the lectures, and much research I think I’m inching closer to grasping the concept.

What I’m gathering is that we implemented the Role class using composition. The Actor will be destroyed when the Role is destroyed. With CastList, I am leaving the role to get created independently of the Cast List. This I believe to be aggregation. Funnily enough, I initially created it the other way, but I really wasn’t sure which was the best approach. I ended up refactoring and cleaning up some of my code and decided to go this route instead. However, if I understand the purpose of the cast list is to just print the full list of roles or to provide an artifact of the full cast, then I would say that aggregation is what we should use.