READ.ME – Daniel Gallagher – G00360986

For my design I used a delegate class combined with an iterator and containers. By using Kappa as a container for the 4 objects I could call them uniformly inside the delegate and grant access to those classes through the delegate inside of Delta and Alpha. This removes the need for Alpha and Delta to know about the underlying structures of Zeta, Epsilon, Gamma and Beta thus encapsulating them. Omega acts as a state manager and iterator for Alpha, Delta and the delegate so that it hides the implementations of Alpha and Delta along with any delegate classes from the client. This allows for greater reusability and extensibility of the code through the Delegate, Omega and Kappa as if we were to make a new object class, we could place it inside the Kappa container and Delegate would not need to change neither Alpha nor Delta.

Overall using this design, I have achieved a structure which has high extensibility, a lack of dependencies and is greatly encapsulated.