**How does your design implement the four pillars of OOP(abstraction, encapsulation, inheritance and composition, and polymorphism)?**

The Arena class holds a number of Character objects that are able to do battle with one another. The 4 Character objects have combat attributes and methods that both display information about those attributes as well as methods to execute attacks on one another.

Abstraction is implemented through methods such as get\_damage() and take\_damage(). Both methods do calculations for combat damage whether its outgoing or incoming making the Arena class’ job of executing fights that much easier.

Encapsulation is implemented through various private attributes within the Character classes. Inheritance is implemented by having our subclasses inherit base attributes and methods from our AbstractCharacter such as get\_health() and get\_username().

Composition is implemented by our Arena class that is an aggregate of Character classes.

Polymorphism is implemented through several of the class’ get\_damage() and take\_damage() methods. Due to some classes having the ability to critically strike, cast a spell, dodge, or use a shield for defence, one or both of these methods are overridden on all classes however the method name remains the same as the parent class’.

**Why are your entity classes good abstractions of the real-world entities?**

Our classes model characters that do battle within an arena. They have damage modifiers and can both inflict and take damage. They are good abstractions of real-world entities because they have attributes and methods you would expect from them. For instance, the MageCharacter has an overridden method get\_damage() that returns damage that has been powered up by its \_spell\_power attribute if it manages to roll a number that is within its \_spell\_cast\_chance. The KnightCharacter has an overridden method take\_damage() that further reduces damage based on his \_shield\_modifier attribute. Finally, they all have a \_health attribute and an \_attack\_speed attribute.