Our design implements abstraction by using an abstract class (abstract\_player) that we created. By doing this, we were able to create different types of players that used abstract\_player allowing us to save time and have cleaner code. In our project we also used encapsulation. This allowed us to restrict access to variables which prevented data from accidentally being modified. We also used inheritance and composition to allow our classes to inherit methods allowing us to save time and keep our code clean. As for polymorphism, we used get\_description between both Abstract player and guard, forward, and center which again, helped to make our coding more efficient.

Our entity classes are good abstractions of real-world entities as we took a complicated real-world thing which is a basketball player and we simplified it into a class which contains only the most important and relevant things. Basketball players have important statistics that represent their skills in the game and so we took the most important statistics to a player such as number of shots made, the players name, the players height, etc. and added them to the class.