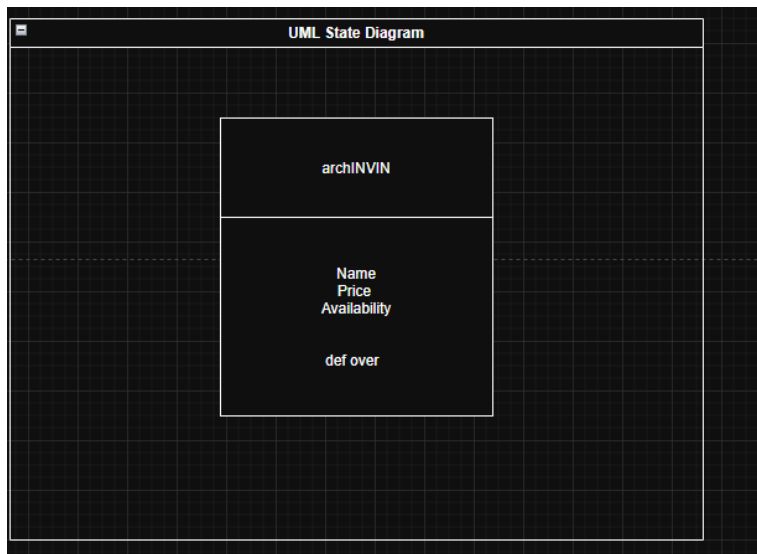


# Object Oriented Analysis

This report will focus on the differences between python and C++ that were used for the creation of the e-commerce app for a sporting goods store. The app was structured by using Object Oriented Programming concepts to accomplish a seamless browsing that cataloged each product with essential information. Here's a UML class diagram of the parent class archINVIN used in



the app below. Each item was categorized by taking the attributes that were shared with other items and then handling the unique attributes in their respective class. Starting off, I believe python easily took the category for its ease of implementation. Python ended up

being easier to write in complexity and was shorter in terms of lines of code. Python is known for its simple and intuitive syntax, like its ease of use to print something. I found the syntax in implementing OOP in C++ to be more confusing and the data structures like sets in python were easier as well. Furthermore on the OOP aspect, implementing it in python was easier as it was very self explanatory, as the line to call the overarching class archINVIN was intuitive as shown

```
def __init__(self, name, price, availability):  
  
super().__init__(name, price, availability)
```

Compared to C++

```
archINVIN(string name , double price, int availability) : name(name), price(price), availability(availability) {}  
Volleyball(string name, double price, int availability, string material, string VolleyBallsize) : archINVIN(name,  
price, availability), material(material), VolleyBallsize(VolleyBallsize) {}
```

As shown above, this demonstrates how Python outpaced C++ regarding user-friendliness and ease. Next, I believe python is tied with C++ in regards to code readability and maintainability. C++ has explicit syntax that makes it harder to read but it leads to having less doubt about what's happening in the code. The explicit typing gives a lot of useful information at a glance which makes it easier to maintain. Though, Python is easier to read, more flexible and there is less code to look at. Python is known for its easeability to read and write code which makes it where I believe there isn't a clear cut answer to which language is easier to read and maintain. The two languages definitely deferred when it came to handling object oriented concepts. I found python to be overall easier to write compared to C++. It was easier to utilize encapsulation in python, as I did not need to set information in the classes to public, private or protracted like you would in C++ . It also was easier to implement polymorphism in python, as it was straightforward and efficient when coding to utilize functions in my parent class in the other classes. Each class was noticeably easier and shorter to code in python then C++. In C++ I had made the function Over, used to display information about the items, virtual in order to be used by other classes. Finally, error handling in python and C++ are handled in similar fashion. I didn't have to use them in the e-commerce app but I have used them in the past. They both use exception blocks that take an argument that if met will output an error message. In a border sense, I find it easier to debug python as in run-time it gives better information on what went wrong. There are differences that went unnoticed by me for certain, but I found making the e-commerce app in python to be a better experience.

