

While working on this for the sports inventory systems in both Python and C++, the implementation for Python could just use one file to help direct the code, while C++ can implement multiple files into the system. The ease of implementation is shown with Python regarding this, while with C++ I had to plan out a bit more with the code itself, for instance, with creating header files for the main class to pull for the desired class pull. Python does not require these header files since all classes can be implemented into a single class type and still come out with an output for it to work. Python's code is also more readable due to a clean and concise syntax base. Since Python doesn't require a type declaration or header files, the direction of the code could be followed more simply by reading down the code. With C++, the files can be jumbled between each other, contracting the readability and creating a file pull base flow instead of a line base flow. These instances of readability can be helpful in some sense with the code languages having different strengths, with python having a ease of implementation and readability, but C++ having a pull-based system, which can help with edits needed later when implementing new updates to a project file type. Encapsulation in python can be achieved through using underscores, while C++ uses access specifiers like private, protected, and public. Inheritance is allowed in both classes, with Python having a simpler and minimal syntax, while C++ the inheritance must be explicitly declared with access modifiers which adds clarity but increases complexity. Errors in both Python and C++ can be caught while implementing the codes via the terminal. The main error I ran into while creating this code was creating runtime errors because of constant print issue I created by overcomplicating the code itself. C++ ran built had incorrect inheritance due to issues with misnaming the files by one letter. Both Python and C++ can successfully implement OOP inventory system, but the experience is vastly different. I spent more time creating and implementing plans for the C++ than Python since python is ideal for quick development in comparison to C++.. For these differences in the development and planning phases, one required a shorter implementation and debugging time than the other.