

This assignment was actually fairly easy, as I had accidentally done the last assignment using polymorphism, so it was just a matter of using the same techniques, and applying them to different classes, with different types of data. To start out, I knew I would need a parent class with a few shared pieces of information. I also knew that I would need to declare the functions of the subclasses as virtual functions of the parent class. I did not at the beginning know how I would do functions only used by one subclass, but I was able to figure that out. The main issues I was getting when I tried to compile were linking everything together. I was admittedly lazy and didn't want to do header files, so I started with none but then I was getting redefine errors so I decided to go ahead and use some. After that it was making sure they were included in the correct places and all that.

Creating the subclasses was also fairly simple. Just using rands and a somewhat standard structure between the different subclasses for the attack and defend functions meant I could pretty much copy and paste, though they were just different enough that I wasn't able to make a function in the parent class for them. Then doing some specific stuff like for the blue men, and the unicorn took a bit of thinking. Baba yaga was the only child that required a function other than the attack and defend, so I had to figure out how to implement that class. To do this I had to declare it as a non-pure virtual function in the creature class and it seemed to work out fine.

As far as error checking goes, there weren't many user inputs, but I'm pretty sure that they all check for errors, only liking certain char values, and anything else spitting out a bad input error. The main program is fairly bare bones, just meant to test the classes and monsters. I did remember at the last second to clear the memory though and running it through valgrind, it shows it deallocates all memory used. Formatting for the output was also fairly bare bones, just spitting out the monster's ID, a character, and its health for that turn.

As far as changes from my original design go, I didn't stray too far from it. Like I said this assignment was fairly straight forward, so I just had to write it all out and that was it. The way I did header files is a bit wonky, but it does work, and I probably didn't even need some of them. I was getting errors with files being defined multiple times and I realized I had forgotten about the `#ifdef` and `#def` stuff at the beginning of the header files, so I did that and it solved a lot of my issues. I also remembered this time to do function descriptions, since I manage to get docked for those because I don't like doing them I also attempted to keep all functions fairly small, using others where there was repetition.