

## New Egg

I start off my program with a base class called products that has general fields that most products that are sold on the online store new egg. From the CPU, manufacturer, and memory. I have getters and setters and a constructor so its basic at the start, a general class so my subclasses can inherit the general methods and fields and add other methods to make that subclass unique. After my product base class was set up, I made three subclasses which were basic products like laptops, desktops, and monitors. Each of those subclasses had different methods and fields to make them unique. I used basic inheritance for that part of my code to showcase my coding skills in Java and what I learned about inheritance. After I created those four classes I moved onto my Factory class that used a Singleton and Factory pattern that we learned in class. I used some of the slide decks that were provided and few tutorial point examples to work around. I basically made an array list to store my products from my previous class and made a static factory field then populated my factory constructor with products, for example, the monitor, desktop, and laptop. I used getters and setters after, also I had created objects from my laptop, monitor, and desktop class to populate my array list. This class was mainly used to use the two Java pattern and use an array list to use polymorphism later in my program while creating objects. Then I moved on to my generic class which accepted to types T and U which were matching the class name internet security, I just made a simple constructor and a to String to go along with it. After that generic class was created, I moved onto using a class I made called gaming chair which was used in another class called game setup which used a has-a relationship instead of inheritance. Last I went to my testing aka main class to test the program. All I did for this was create an object for game setup and my generic then printed out the outputs for each, while using a for enhancing loop to print out all the products which showed overriding from the subclass to String method instead of the base class Factory to String method, showing polymorphism. I did not go with enums due to the fact, I couldn't really do much with it except for showing different categories and I did not seem it was necessary.