Raphael Genova
CS342
Hallenbeck
4/23/20

Template Design Pattern

Description of use in the program: The way I used the Template design pattern in my program was I made an abstract class called Person which is the baseplate for the two classes that inherit from Person called Hero and Villain. The Person class had 2 final void methods called: intro() and showcase(), and 4 abstract methods called: createPerson(), ultimateMove(), sidekick(), and headquarters(). The final void method, intro(), has a try catch block of trying to print out the introduction of the person with their codename, secret identity and quirk, if the try block fails, then catch throws an exception and prints out the secret identity of the person. The other final void method, showcase(), calls all of the abstract methods inside the Person class and displays them accordingly. The Hero and Villain classes, as mentioned before, inherit from the Person class and they both use it as a baseplate for what they are going to become because that is how the Template design pattern is used.

Pros of the Template Design Pattern: A major pro of the Template design pattern is that it's very easy to use. This is a pro because other design patterns for java get really complex and hard to conceptualize.Other design patterns just make your head spin with all the terminology and certain different constraints. Which leads into the next pro of this design pattern is that it's inheritance. Inheritance is very easy to conceptualize and understand because it's one of the first things taught when learning Java. This concept is easy to write down and draw out on paper by using a UML diagram or just chicken scratch.

Con of the Template Design Pattern: A con of the Template design pattern is that it gets repetitive. This is a con because each class that inherits the parent class must conform to the design and functionality of the parent class.