For Assignment five we were tasked with updating our last three assignments. Using the factory design pattern for the creation of the shapes and using a design pattern of our choice for the styling of the shapes. The factory design pattern was the easiest design pattern we have learned this far. the main thing the factory method is used for is to help with the creation of objects. For example, say you wanted to code a vehicle making company and first you start out with cars and then you want to add trucks, motorcycles, and vans. Instead of creating each one in the main program you could just call your vehicle factory method and it will handle the creation of all the different objects. So, as you can imagine, this greatly improved my code and made it very easy to read. Implementing the factory design pattern made my main program very small in all three of my past assignments. I was able to add the factory method into each of my assignments by first creating a new ShapeFactory class. I first started with assignment 4 and I was able to just take my switch statement I had in my main method and put it into my ShapeFactory class. Then in my main program all I had to do was call my shape factory when the user wanted to add a new shape. I then repeated this in all three of my assignments with only a few minor adjustments. I also thought it would be nice for the user to pick between a default shape or a custom shape. So, I made another method in my ShapeFactory and put an if else statement in my main program for this to work. Once I had my ShapeFactory working in all three assignments I moved onto the styling for the shapes. After talking to a few friends and demonstrators asking them what design patterns they thought would/could be useful for this part of the assignment. I got two answers use the singleton design pattern or another factory design pattern. After looking into the singleton pattern, I decided to do another factory design instead of a singleton pattern. I learned that the singleton pattern is not very commonly used in the industry because it is a global variable. It is best to stay away from global variables because they could very easily be accidentally changed somewhere else in your code as they can be accessed from anywhere. That was why I choose to do another factory design for the styling. So I made a new file called StylesFactory in my assignment four to start with where I made a method that asked the user what styles they want. I had my ShapeFactory call my Styles factory so that everything could be done at once. I also had another method for default styles in case the user didn’t want to worry about the styling of the shape they were creating. so I added if else statements in my ShapeFactory asking the user if they wanted custom styles or not which then called the StylesFactory. Finally, I did this same thing for each of the three assignments and I was finished. They only downside to doing another factory method for my styles was that it made my ShapeFactory very long. However, I think it works well for this assignment.

Assignment 4 UML

Diagram

Description automatically generated

Assignment 3 UML

Diagram

Description automatically generated

Assignment 2 UML

