For Assignment 3 and 4 we were tasked on creating two design patterns, Memento and the Command design pattern. Based on two of the Assignments I feel like the Memento design pattern worked best for me in my solutions for both projects for numerous reasons. I felt it worked better because for both assignment 3 and 4 the console application wasn't as large of a project for using the Command design pattern where this design pattern is most affective with. Where as for Memento although there was a lot of code written it was mostly repetition (an example see whenever I wanted to save something for memento I would have to repeat the same line for saving memento for every shape I added into my canvas whereas in the Command design pattern I save and access it inside my invoker class) and this is why I felt Memento worked best with. Using the Command design pattern is best for very large and scalable projects. If I were to add more commands into this design pattern this system would still be as efficient because instead of repeating a lot of code we can just create an instance of an interface for the command and call this command which would keep it scalable. Referring to my Assignment 4 all I would have to do is just create an instance of my command interface in a different class to create a command and then add it to my invoker class to then execute it in my main method, instead of creating a method and when using this command saving it whenever I use it in my program in my Memento class. For both assignments both assignments 3 and 4 I ran into trouble just understanding the design pattern firstly then once understanding them I ran into trouble working out how to implement each design pattern based on the tasks for each assignment. I solved these problems by firstly watching YouTube videos on each design pattern until I could vocally describe myself. I then got examples online or was given code from lecturer and then ran the code and followed through the program(main method) while looking at the console to understand what exactly each line was doing and if there was classes called I would go back to the class and see what was done in the class. Once I had done this I would try and implement the tasks required in the assignments while following alongside the example code from online or the lecturer. Another problem I ran into was with the format of the svg when saving into a file. For there is usually a format to follow to make a svg file there's code you need to put at the start and at the end. But for adding anything into my canvas stack the variable i had was shape variable not a string variable(Stack<Shape> canvas…) which wouldn't allow me to add these lines at the start and at the end. So what I did was created a "Start" and "End" Shape class amongst my shape classes. Inside instead of adding a shape I added the start and end of the svg file format in each one then used those classes to add them to the stack.