**Assignment 04 Report**

The undo-redo functionality for the memento design pattern and command design pattern are very different. For my memento design pattern, I had to create both a memento and caretaker class for the design pattern to function properly. I also made my canvas a list to make it easier for adding and removing shapes for my canvas for the undo-redo functionality. For my command design pattern, I had to create a user class and a command abstract class. This was done as I would then create an addshapecommand class which inherited the abstract classes’ attributes. I also instead of making my canvas a list, I made it an actual an actual class, to store the different canvas methods. These were all created for the command design pattern to function properly.

When designing these 2 patterns I found the memento pattern was easier for my solution. I found this because I felt more comfortable using it than the command pattern. I understood its functionality more and felt that I could implement it better. For example, because I had made my canvas a list rather than using it as its own class and storing it as a stack in that class like command I found adding and removing elements for it easier for my undo-redo functionality. I also understood the role of the caretaker better than the role of the user in my command pattern. The caretaker stored my memento and its states. Whilst I understood that the User class was the invoker class for the pattern and the canvas was the receiver, I didn’t fully understand how to implement certain functionality I was able to in my memento program.

For example, in my memento pattern I was able to have it, so my program knew when my canvas was empty when utilising the undo or redo functionality. I then would print a message saying, “Canvas is empty”. I couldn’t get this functionality working with my command pattern as I didn’t make my canvas a list so it would just display a blank line in the terminal if the canvas was empty. For the command design pattern I also had to create execute (do()) and unexecute (undo()) in my command abstract class which was something I didn’t have to do in memento. This was done so the undo-redo functionality would work correctly.