Greg Hatfield

Write a one-page journal entry describing all four different styles and patterns you used.

What were their strengths and weaknesses?

Which is most appropriate for this problem?Why?

The four different styles that were used in these bowling projects are procedural, object oriented, pipes, and blackboard. These styles solve the same problem but with different patterns and approaches. I found that these styles can get the job done with the bowling assignments but can make the assignment more complicated than it needs to be.

The procedural pattern just goes to the next line and is done all in the main program. No functions or classes are used. This can work the bowling program but it can get quite messy with multiple if statements and loops. Especially when dealing with multiple players. Which will not make it ideal to do this kind of assignment when working with large number of games.

The object oriented pattern creates objects and makes use of functions. This makes keeping track of multiple players and the game quite stably. This pattern will be the best to use for the bowling assignments especially when dealing with multiple players. However the object oriented method must be well designed so you can change the classes and functions when needed otherwise you have a hard time maintaining the code and have to do more work than needed.

The pipe and filters pattern reads from multiple files to get the data it needs to output the final score from the player. For multiple players, it can be just as easy since you can deal with the object oriented classes and functions. For bowling it seems like overkill and currently I only know how to work this pattern out with using console based applications. If the bowling has to be moved a form instead of console, I don’t think pipe and filters will work as well.

The final pattern was the blackboard style. This style was difficult for me to understand what it does exactly. From what I understand it is a class that is fed in data and it makes a decision based on the data being fed in if the program is done or what the current score is. I was not sure exactly on how to implemented It and got quite confused. This program will work great when the fate of the game is unknown and it is being fed constant data from multiple players and games. I don’t believe this would work best for the bowling game since majority of bowling games is pretty procedural in the data being fed in. So random calls won’t be needed.

Over all, object oriented pattern is the strong choice for these kinds of programs. Because it is easier to work with objects whenever you want rather than being called randomly, being depended on other programs, or just straight up procedural run it. It works better at your pace and based on the design and how maintainability it has. Can be used for a wide range more programs.