Aidan Weber-Concannon 260708481

- The Strategy pattern allows you to choose how your program will react at run time, while the template's implementation must be chosen at compile time.
- In our code template is easier for user to use, the implementation is simpler and it requires less coding in the driver class.
- The Strategy pattern requires more coding on the user's part.
- The Strategy pattern requires editing of the interface, and original class to add additional functionalities.
- Template pattern requires editing of the abstract class to add more functionalities, this is not considered good practice.
- The broadness of the strategy design makes it easier to add more strategies in the future without having to edit the original code.
- The template design pattern provides a simpler user implementation and less reuse of code but this comes at the cost of decreased flexibility when adding additional functionalities.
- The template design is more robust to corruption as it can error check in the abstract class and leaves less decisions to the user.
- In general, the strategy design gives a lot more flexibility in the user's solution to a problem, while the template provides a more structured solution.