

- Examine your code and identify opportunities for code improvement. Explain the specific opportunities and the reasoning behind them
 - Implemented decorator design pattern to improve code reusability, reduce complexity and effortless extensibility.
 - Code Reusability - By using abstract classes for beverages and beverage add-ons, new beverages and add-ons can be added easily.
 - Reduce Complexity - Previous implementation had a two tier inheritance which was tightly coupled, with the updated implementation beverages and add-ons have been loose coupled so reduces complexity under separation of concerns.
 - Effortless Extensibility - Previous implementation condiments were an attribute inside the Beverage class so new condiments cannot be easily added, with new implementations they are separate classes so new condiments can be easily added.
- Describe the changes made to the code. Document the details of the changes and create a UML class diagram that illustrates the implementation of the design pattern(s) used

