# HW5 – Putting Principles, Practices, and Patterns to Work

*Estimated time: 12-20 hours*

## Objectives

- Use the principles, practices, and patterns learned thus far to improve an existing system
- Learn to recognize and avoid common pitfalls

## Overview

For this assignment, you will select one of the systems that you built for one of the four previous homework assignments and then improve the quality of that system by

- Improving localization of design designs, encapsulation, abstractions, coupling, cohesion, dependency inversion, modularization, maintainability, extensibility, or reuse
- Applying design patterns not previously used in that system
- Adding more thorough and useful test cases

You may extend the functional of the system, but don't get carried away. The primary purpose of the assignment is to improve the quality of the old system in meaningful and defendable ways.

## Instructions

1. Decide which of the previous four systems you want to improve
2. Review your old design and implementation and then identify areas that need improvement. Look for common pitfalls that you may have fallen into.
3. Make use of any design pattern that can help you improve the system
4. Don't just use a design pattern just for the sake of using something different. Its use must be justified.
5. Look for interesting opportunities to extend the functionality of the system without dramatically increasing its complexity.
6. Redo your design document (UML Class, Interaction, and State Chart diagrams)
7. Implement your changes
8. Test non-GUI components thoroughly
9. Write up a short summary (less than 1 page) in a README file that explains your improvements and enhancements, plus your rationale.

## Submission Instructions

Zip up your entire solution, including test cases and sample input files, in an archive file called CS5700_hw5_<*fullname*>.zip, where fullname is your first and last names.  Then, submit the zip file to the Canvas system.

## Grading Criteria

| **Criteria** | **Max Points** |
|---|---|
| Also, a README file that describes your improvements and rationale | 20 |
| A clear and concise design document, consisting of UML class, interaction, and state-chart diagrams, that reflect good design principles | 20 |
| A quality implementation reflects good OO Practices and avoids pitfalls | 30 |
| Thorough executable unit test cases for non GUI components | 30 |