

CMPE 202

Software Systems Engineering

Section 47

Spring 2026
Instructor: Ron Mak

Assignment #4

Assigned: Wednesday, February 25
Due: Wednesday, March 4 at 11:59 PM
Team assignment, 100 points max

Design Specification

Write a Design Specification for an imagined project. Be creative – at the end of the semester, we won't hold you to this project. Your software design can use your classes from Assignment #1, or you can do something completely different. A design spec should be read and understood by the software developers.

Your specification should include:

- **Well-design classes** (at least four)
- **UML class diagrams** for your important classes. Show the relationships between classes using the appropriate connectors. Show any multiplicity. Include some important attributes (member variables) and methods (member functions).
- **Describe your good class design** by pointing out how your classes are cohesive and loosely coupled with hidden implementations. Discuss how you used aggregations and/or compositions.
- **Encapsulation.** Discuss what can change in your application and how you encapsulated those potential changes.

You can use a UML drawing tool to create the diagrams and insert the diagrams into your specification. Two free UML drawing tools:

- Violet: <http://horstmann.com/violet/>
- StarUML: <https://staruml.io/>

Use your imagination! You will not be asked to write a program that implements everything you put in this Design Specification.

What to turn in

Each team should create a PDF containing the Design Specification. Name the file after your team, such as **Supercoders .pdf**. Submit it into Canvas: **Assignment #2:**

Design Specification

This is a team assignment. Each member of the team will receive the same score.

Rubric

Your Design Specification will be graded according to these criteria:

Criteria	Max points
Well-designed classes (at least 4) <ul style="list-style-type: none">• Good names• Well-named member variables• Well-named member functions	30 <ul style="list-style-type: none">• 10• 10• 10
UML class diagrams <ul style="list-style-type: none">• Correctly drawn class diagrams• Good class relationships (dependency, aggregation, inheritance)	30 <ul style="list-style-type: none">• 15• 15
Descriptions of how your classes are: <ul style="list-style-type: none">• Cohesive (single responsibility)• Loosely coupled (minimal dependencies)• Hidden implementations (public vs. private)• Encapsulate change	40 <ul style="list-style-type: none">• 10• 10• 10• 10