

CSC 319 Object Class Design

Remembering the four principles:

1. Encapsulation
2. Polymorphism
3. Composition
4. Inheritance

Design a Class

Here are the high-level specifications for your program and class.

You're creating a program that allows a student to create their own transcript.

Specifications:

- Store student name
- Allow students to create semesters. We number semesters as a year and extension. This is 202540. The extensions are 10 (winter), 20 (spring), 30 (summer), 40 (fall).
- Each semester contains a list of courses with the course designation (like CSC 319), course name, number of credits, and grade.
- User interface allows creating and deleting semesters.
- User interface allows creating and deleting courses within semesters.
- User interface allows displaying a semester.
- Sort courses by course number or grade.
- Summary display that shows semesters, credits, and GPA.
- Calculates GPA by semester and cumulative.
- If a course is retaken, the last grade counts for cumulative GPA.
- Store the data into a CSV file.
- Allow reading in data from a CSV file.

Part 1:

- Design a class that holds the data and has method definitions (not code)
- Figure out what methods and attributes you need.
- Document these in UML.
- Feel free to use Mermaid (mermaid.live). See some hints in Brightspace.

Part 2:

- Design the class (don't implement the code at this point).
- Just the MyBanner.hpp file.

Part 3 (Homework project):

- Implement your class

Part 4 (Add on to the homework):

- Add "modeling scenarios"
- Allow future classes to have "scenarios" that you name.
- In each scenario you can have different grades for future classes.
- Write out a report that shows the GPA for each future scenario.

Hints:

1. Designing your class:
 - a. Start with your attributes.
 - b. What do you need to store? How do you need to access it (by name, by grade, by semester?) This helps you decide what data structures to use.
 - c. Convert commands into methods.
 - d. Specify support methods (methods that do small pieces of work)
2. Define your command inputs.
 - a. Write these down in a document.
3. Some rules of thumb say 20-50 lines of code for one method
 - a. Think of about ½ page or so of code.