Clifford Armstea lll

October 20,2025

Midterm

1 A) High cohesion. A well-designed class should keep all its responsibilities closely related to one intent/abstraction this improves comprehensibility, testability, and change locality.One reason to change.

B.Low cohesion (violates Riel’s one key abstraction heuristic). StudentPortalHelper mixes unrelated concerns GPA calc (domain), CSV export , email text & date formatting (presentation), payment processing (integration), password policy (security), and a global cache (infrastructure).

Refactor: split into focused classes. GpaCalculator, RosterExporter (+ CsvWriter), WelcomeEmailBuilder, UiDateFormatter (or keep in UI), PasswordPolicy, PaymentService (+ PaymentGateway port), and InMemoryCache (no static). Use dependency injection; name by intent, not “Helper.”

3 A.) No. If trim is modeled via inheritance ( BaseCar, LuxuryCar, SportCar) or via immutable/final fields, the trim is baked into the object’s type at construction time. Changing trim would require creating a new subclass instance, copying state, and updating references breaking identity and making persistence/workflow error-prone.

B.)Refactor with composition (Strategy). Make Car compose a Trim and an Engine role:

class Car { VIN vin; Trim trim; Engine engine; void changeTrim(Trim t){ this.trim=t; } }

Define Trim (BaseTrim, LuxuryTrim, SportTrim) and Engine (ElectricEngine, PetrolEngine) as interfaces/classes that encapsulate trim/engine behaviors (price deltas, features, specs). Car delegates to them and can swap implementations at runtime, preserving identity (VIN) while allowing dynamic trim changes.

5.) Throughout this course, I’ve used AI tools like ChatGPT to help me better understand topics such as object oriented design, Java syntax, and how to organize my assignments. When working on labs or projects, I often turned to AI for help clarifying UML diagrams, checking if my class relationships made sense, or reviewing my code for structure and readability. Sometimes I also used it just to get a clearer explanation of a difficult concept or to double check that my own explanations were accurate before submitting my work. Having that kind of quick, on-demand support made learning feel smoother and less stressful, especially when I couldn’t get immediate help from my instructor.

The biggest benefit was how fast AI could give feedback and simplify hard ideas it helped me learn by showing examples and explaining why something worked. A limitation, though, was that it sometimes gave answers that were too general or slightly off, so I had to fact check everything against what we learned in class. Going forward, I expect AI to keep playing a big role in how I solve problems, both in school and in my career. It’s a great brainstorming and debugging partner that helps me think through logic and improve my designs, but I’ve also learned to balance it with my own judgment to make sure the final work is accurate and original.