Large Scale Programming: Midterm Essay Questions

**1a.)** A well-designed class should have high cohesion. High cohesion means that all of the class’s methods and data members work together to fulfill a single, focused purpose. This improves readability, maintainability, and reusability, because developers can understand what the class is supposed to do at first glance. Low cohesion, however, mixes unrelated functionalities which makes the code harder to debug, test, and extend.

**1b.)** The StudentPortalHelper class has low cohesion because it bundles together many unrelated responsibilities. The following methods serve a completely different domain concern, violating Arthur Riel’s heuristic that a class should represent one and only one concept;

* Academic logic (computeGPA)
* File I/O (exportRosterToCsv)
* Email generation (makeWelcomeEmail)
* UI formatting (formatDateForUi)
* Payment processing (processTuitionPayment)
* Security (isStrongPassword)
* Caching (putCache, getCache)

To improve cohesion, I would split the class into smaller, single-purpose classes like so:

* GPAService – handles grade averaging and GPA computation.
* FileExporter – handles roster export.
* EmailFormatter – formats email templates.
* DateFormatter – formats dates for UI.
* PaymentProcessor – handles tuition payments.
* PasswordValidator – checks password strength.
* CacheManager – manages key-value caching.

Each of these classes would have clear responsibilities and could be reused independently, increasing modularity and testability.

**3a.)** No, the current design does not support dynamic trim-level changes. The UML structure shows Car subclasses (Base, Luxury, Sports), meaning trim level is fixed at instantiation (compile-time binding). Once a Car object is created as a Base or Luxury car, its type, and therefore its trim, cannot change because Java does not allow changing an object’s class at runtime. This violates the Open/Closed Principle and makes the system rigid to customization during the manufacturing process.

**3b.)** To allow dynamic trim changes, Car should use composition instead of inheritance for trim levels as shown:

class Car {

private TrimLevel trim;

private Engine engine;

public Car(TrimLevel trim, Engine engine) {

this.trim = trim;

this.engine = engine;

}

public void setTrimLevel(TrimLevel newTrim) {

this.trim = newTrim;

}

public TrimLevel getTrimLevel() {

return trim;

}

}

5.) Before taking this course, I mainly used AI tools like ChatGPT to help me understand programming concepts that I found confusing or unclear in class. During this course, I relied on AI more interactively — for example, using ChatGPT to review code structure, explain Java syntax, and clarify how abstract classes and interfaces work. It helped me reason through programming problems rather than just giving me the answers, which improved my ability to debug and organize my code logically.

One major benefit of using AI was how quickly I could get targeted explanations and feedback while coding. It allowed me to test ideas and understand my mistakes. The main limitation, however, was that AI sometimes gave answers that were too general or didn’t match my exact assignment setup. This taught me to double-check everything and not rely on AI blindly.

Going forward, I expect AI to continue being a learning companion in my academic and professional work. I plan to use it as a brainstorming and debugging tool rather than a shortcut. It will help me refine my understanding, document code, and learn new technologies faster. I believe AI will become an essential part not just of modern programming, but in the modern workplace in general, especially for improving productivity and creativity when solving complex problems.

**Citations:**

[**https://chatgpt.com/s/t\_68f6bf029a6881918a934ac2511cc0bc**](https://chatgpt.com/s/t_68f6bf029a6881918a934ac2511cc0bc)

[**https://chatgpt.com/s/t\_68f6bf47c028819190983dc6aa917762**](https://chatgpt.com/s/t_68f6bf47c028819190983dc6aa917762)

[**https://chatgpt.com/s/t\_68f6bf7d4a148191a20679917b63a817**](https://chatgpt.com/s/t_68f6bf7d4a148191a20679917b63a817)

[**https://chatgpt.com/share/68f6c95d-ac54-8002-8160-d9623693675f**](https://chatgpt.com/share/68f6c95d-ac54-8002-8160-d9623693675f)