

Review Quiz Questions

DANGER

Note that the followings are **Sample** questions only and their purpose is to provide some practice with the concepts/material covered in the class. The following is NOT a representative of the actual exam by any means. The actual exam may (and probably will) be different in terms of level of difficulty, types of questions, length, format, content etc. In general, if you have studied and have a deep understanding of the material taught in the class throughout the semester, you should have no problem in the real quiz!

True/False

Please indicate if each statement is `true` or `false` .

Statements

1. **Continuous Integration** is a software testing technique aimed to expose faults in the *interaction* between integrated units ; it must be done at the end of each agile (development) iteration.
2. The **dependency inversion principle** in object-oriented design is always being used if we define and use interfaces.
3. In Object-Oriented programming (OOP), the primary purpose of "polymorphism" is to reuse your code.
4. Coupling is a measure of how strongly one element in code (such as a class or a method) is *focused* and comprise of responsibilities which belong together.
5. Open-closed principle states when extending a class, consider that you should be able to pass objects of the subclass in place of objects of the parent class without breaking the client code.
6. Adapter is a structural design pattern that allows objects with *incompatible* interfaces to collaborate

Multiple-Choices

Please indicate the correct choice (there is only one for each question).

Questions

1. In software development, "**requirements**" are about:

- ☐ a. what the software system should do
- ☐ b. how the software system should work
- ☐ c. both about what the software system should do and how the system should work
- ☐ d. none of the above

2. When the **REST principle** is being used to describe the interactions between a *client* and a *server*:

- ☐ a. the **server** maintains the state of the client
- ☐ b. the client can send different kinds of http requests: *create, get, put, post* and *delete*
- ☐ c. the **client** maintains state
- ☐ d. both the client and the server maintain state

3. When using an agile development process, the **product backlog**:

- ☐ a. is a list of the products a company needs to build
- ☐ b. is a prioritized list of backlog items specific to the product under development
- ☐ c. is a list of defect associated with the current product
- ☐ d. is a list of engineering tasks negotiated by the team and the customer about the current development project

4. Which SOLID design principle is most closely described by the given statement: **classes should depend on abstractions rather than implementations.**

- ☐ a. Single Responsibility Principle
- ☐ b. Open/Closed Principle.
- ☐ c. Liskov Substitution Principle.

- ☐ d. Interface Segregation Principle
- ☐ e. Dependency Inversion Principle

5. DRY design principle is associated with which code smell?

- ☐ a. code duplicates
- ☐ b. data clumps
- ☐ c. primitive obsession
- ☐ d. a and b
- ☐ e. none

6. Which statement is always true about XSS?

- ☐ a. It is a server-side attack
- ☐ b. It is an injection attack similar to SQL injection
- ☐ c. a and b
- ☐ d. attacker's malicious code is stored in server-side before being sent and getting executed in the victims' browsers.
- ☐ e. none

Analysis and Design 1

Imagine you have been asked to implement a **Piazza-like** system (Q/A and discussion for courses). One of your team members defines the following User Story:

As a professor, I would like to be able to post announcements so that they appear on my students' feed.

Part 1

You are implementing this User Story; what classes will be in your "model" UML class diagram? (Only name the classes .)

Part 2

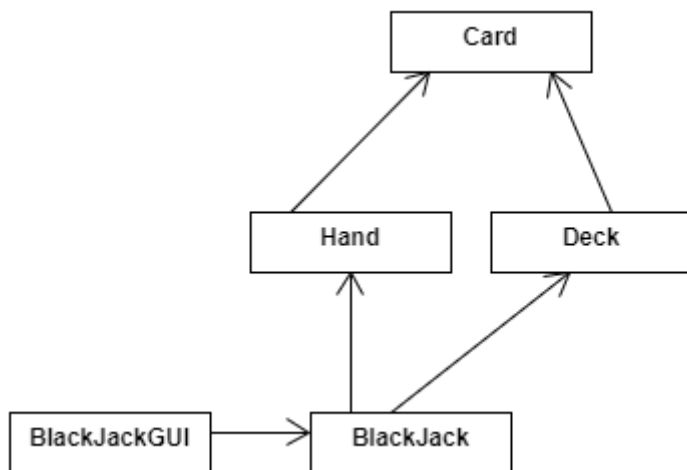
The software system is implemented according to the Client-Server Architecture. To show your understanding of this architecture, describe how the above-stated User Story carries out through the interaction between different entities (user, client, server, database, ...) .

Part 3

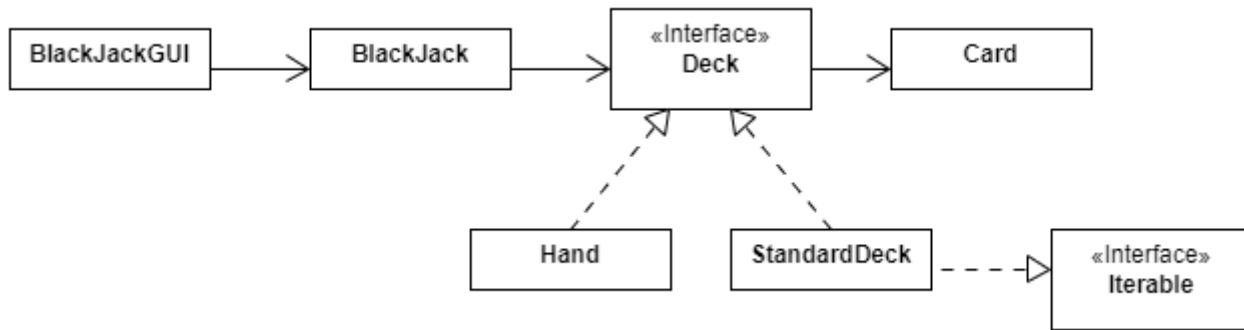
Based on the above-stated User Story (and your general understanding of how a Piazza-like system works), what design pattern(s) [among those we covered in lecture/readings] would be primed for application here. **Name** only one **and elaborate (briefly) on the underlying problem and the proposed solution by the pattern (relate that to how the pattern fits here)**.

Analysis and Design 2

A team of students are building card game application. They are considering to implement the game of Blackjack in their first two iterations. Their UML design includes the following:



The team advisor suggested this alternative design:



Part 1

Briefly explain what SOLID design principle(s) the advisor's revised design adheres to.

Part 2

Briefly explain what design patterns (among those we've learned in this course) the advisor's revised design adheres to.

Part 3

You are asked to criticize the advisor's revised design; what would you suggest?

Design and Analysis 3

Suppose you are designing a software application that will allow the users to perform task management. The user can add tasks to the system and can group tasks together into projects. Projects can be added as sub-projects of other projects, nested arbitrarily deep. Each task has an estimated time for completion that is specified when the task is constructed. You want to be able to treat individual tasks and projects in the same way. In particular, you want to be able to get the time needed to complete a task or a project. The time taken to complete a project is the total time needed to complete all the tasks in the project or in sub-projects of that project.

Part 1

Based on the software description, write two "must have" (functional) requirement in form of User Stories.

Part 2

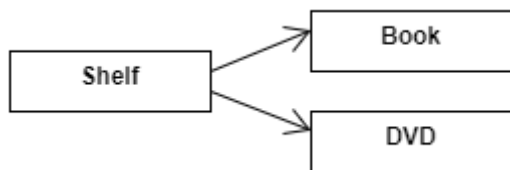
This application conforms to the Client-Server software architecture. To show your understanding of this architecture, describe one use-case (a scenario involving a user using the proposed software) and indicate the interaction between different entities (user, client, server, database, ...) involved in the use-case.

Part 3

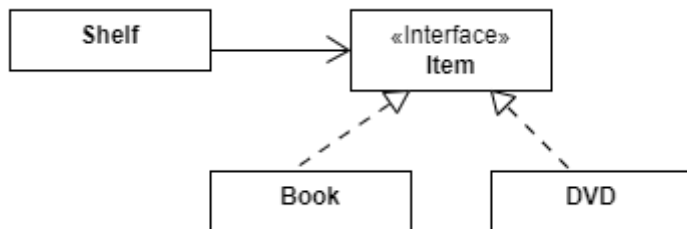
Based on the software description, what design pattern(s) [among those we covered in lecture/readings] would apply to the design of this application. **Name** the design pattern and **elaborate** (briefly) on the problem and proposed solution (how it fits here).

Analysis and Design 4

A team of students are building a Bookstore Management Software. The UML design for the *first iteration* includes the following:



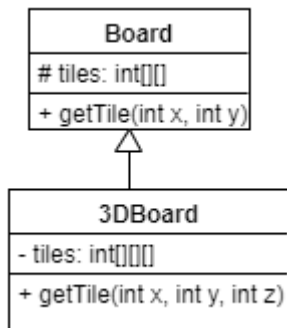
The team advisor suggested this alternative design:



Explain what SOLID design principle(s) the advisor's revised design adheres to.

Analysis and Design 5

Assume you have implemented a board game (e.g. chess) that uses a 2D square-shaped board. You have defined a class `Board` for it that stores all the peices (i.e. tiles) in it in an array. You have also implemented a bunch of methods inside this class including setters and getters, among which there is, of course, `getTile()` which accepts `x` and `y` coordinates and returns the tile that is in `(x, y)` location. Later, the requirements change and now you need to support 3D boards as well. A 3D board *is* a board, so you decide to extend from the existing `Board` class and come up with the following design



Is this a reasonable design? Answer the question by taking into account the SOLID principles. If it is, why? If not, what principle(s) does it go against and how/why? Explain in details. Also, if it is not a sensible design, explain how you would fix/improve it.

 [Edit this page](#)