**Chapter 7**

1. Explain the role of requirements in architectural design. Explain the role of requirements in detailed design. (Pg 130)

* Architectural design: Period during which the high-level overview of the system is developed
* Detailed design: The phase in which the architectural components are decomposed to a much finer level of detail

2. How is aggregation different from inheritance in OO? Provide examples if it helps to show the difference.

* Inheritance: extend the functionality of a class by creating a subclass. Override superclass members in the subclasses to provide new functionality. Make methods abstract/virtual to force subclasses to "fill-in-the-blanks" when the superclass wants a particular interface but is agnostic about its implementation.
* Aggregation: create new functionality by taking other classes and combining them into a new class. Attach a common interface to this new class for interoperability with other code.

3. When we employ the technique of generalization in design, what are we doing, and which part of OO design is closely related to this concept? (pg 132)

Create class hierarchies by recognizing that two or more classes have common properties, and we are identifying a common superclass for them. The part of OO design closely related to this concept is Logical view

4. List two differences between the state transition diagram and the sequence diagram. (pg 150 – 151)

* Sequence diagram: shows the execution of a particular use case for the application and the objects (as in instances of a class) that are involved in carrying out that use case. It could either show a single path, or all of the various paths, through the use case, starting with an actor (user, external system, event) initiating some kind of action.
* State transition diagrams: show the various states that are valid for an object (which could be anything from a method to a class to the system as a whole). That could be a particular class or the system as a whole. This type of diagram shows what actions are valid for a given object, depending on what state it is currently in.

5. Describe three different views used in architectural design. (pg 132)

* Logical view: Represents the object-oriented decomposition of a system – that is, the classes and the relationships among them.
* Process view: Represents the run-time components (processes) and how they communicate with each other.
* Physical architecture view: Represents the mapping of the software to the hardware. This assumes a system that runs on a network of computers and depicts which processes, tasks, and objects are mapped to which nodes.

6. What is the difference between data modeling and logical database design. (pg 142 + 143)

* Data modeling: worry only about the high level design - what tables should exist and the connections between them. In this phase you recognize entities in your model and the relationships between them.
* Logical database design: comes after the data modeling when you explicitly define what the columns in each table are. While writing the logical model, you might also take into consideration the actual database system you're designing for, but only if it affects the design (i.e., if there are no triggers you might want to remove some redundancy column etc.)

7. Describe the difference between low-fidelity and high-fidelity prototyping in the design of the interface. Choose one and give the reasons why you would show the client this prototype. (pg 153)

* Low-fidelity (lo-fi) prototyping: a quick and easy way to translate high-level design concepts into tangible and testable artifacts. The first and most important role of lo-fi prototypes is to check and test functionality rather than the visual appearance of the product.
* High-fidelity (hi-fi) prototypes: appear and function as similar as possible to the actual product that will ship. Teams usually create high-fidelity prototypes when they have a solid understanding of what they are going to build and they need to either test it with real users or get final-design approval from stakeholders.

I’d choose hi-fi prototype to present to the client. This prototype is more suitable to present to client because it has balance between design and functionality, which is what all client want to see. Lo-fi will work but it might give the client the wrong impression of their product.

8. Describe a reference architecture. Provide an example of a reference architecture that is in use today. (pg 138)

A reference architecture in the field of software architecture or enterprise architecture provides a template solution for an architecture for a particular domain.

9. List three of the six constraints placed on a networked-based architecture to produce REST.

* Client-server
* Stateless
* Cacheable

10. In the MVC style, what does model really model? (pg 134)

The model represents the data and does nothing else. The model does NOT depend on the controller or the view.

11. For each of the architectural styles mentioned in this chapter, find one example of a software system that uses it (not mentioned in the chapter). (see next page for additional questions for chapter 8) IT 372 Software Maintenance and Evolution Chapter​ ​8

* Pipes- and -filters: Microsoft Docs
* Event-driven: JavaFX
* Client-server: Banks
* MVC: Web applications
* Layered: Photoshop
* Database-centric: Everything
* Three-tier: LAN server

**Chapter 8**

12. What are the two general characteristics of good design that naturally evolve from requirements? (pg 166)

Consistency and Completeness

13. What is the cyclomatic complexity of the design flow shown in Figure 8.4 (p. 187) where the diamond shapes represent decision branches (if-else or case) and the rectangles are statements? (pg 168)

The cyclomatic complexity using the simple formula of (# of binary branches + 1), the cyclomatic number of Figure 8.4 is (3+1) or 4.

14. Define cohesion in your own words. And what are the levels of cohesion? (pg 172)

* Cohesion means sticking together
* The levels of cohesion are:
  + Coincidental (bad)
  + Logical
  + Temporal
  + Procedural
  + Communicational
  + Sequential
  + Functional (good)

15. Define coupling in your own words. And what are the levels of coupling? (pg 175)

* Coupling means the pairing of two items.
* The levels of coupling are:
  + Content coupling (bad)
  + Common coupling
  + Control coupling
  + Stamp coupling
  + Data coupling (best)

16. What are the six C-K metrics for OO design? (pg 177)

* Weighted methods per class (WMC)
* Depth of the inheritance tree (DIT)
* Number of children (NOC)
* Coupling between object classes (CBO)
* Response for a class (RFC)
* Lack of cohesion in methods (LCOM)

17. What is a depth of inheritance tree (DIT) in C-K metrics, and why may a large DIT be bad for design? (pg 178)

The maximum length from the node to the root of the tree. A high number for depth implies more potential code reuse through inheritance but also higher complexity with a higher probability of errors in the code.

18. In contrast to general design, what is of the most interest in UI design?

The “look and feel” of a website, app, or any other product. The colors, layout, over look to attract users.

19. List four of the eight rules of interface design presented by Shneiderman and Plaisant, and briefly describe each rule that you listed. (pg 181)

* Strive for consistency: Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent commands should be employed throughout.
* Enable frequent users to use shortcuts: As the frequency of use increases, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user.
* Offer informative feedback: For every operator action, there should be some system feedback. For frequent and minor actions, the response can be modest, while for infrequent and major actions, the response should be more substantial.
* Design dialog to yield closure: Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and an indication that the way is clear to prepare for the next group of actions.

20. Deeper​ ​Question.​ Describe the relationship between UI design and (internal) software design? Are they related? How? And do they influence each other?

User Experience Design is the process of development and improvement of quality interaction between a user and all facets of a company. Whereas User Interface Design is responsible for the transference of a brand’s strengths and visual assets to a product’s interface as to best enhance the user’s experience. You need both in order to have a good product.