Chapter: overview

State diagram shows the states of the objects and how those states change. Sequence diagram only shows the message interactions among the objects. 4. Mention and describe three different views used in architectural design. Architectural design is a high level view of the system. It can be described with: logical view : represents the decomposition of a system and how those parts work with each other physical architectural view: represents the mapping of the software to the hardware process view: represents the run-time " processes" and flow of the system. 5.

What is the difference between data modeling and logical data base design? In data modeling we are instructing and showing the Entity-Relationship among the objects. In logical data base design, the E-R diagram is converted to the sets of tables and relationships among the tables. 6. What is the difference between low fidelity prototypes and high fidelity prototypes? Low fidelity prototypes are quick hand drawn screens using markers, post-it notes, post cards, and/or cut out images to represent the screens of the system. Simple mockup sketch of the target product.

High fidelity prototypes are screens created using a tool to design the screens. These screens look and mimic the behavior of the future screens of the system. An example of a tool is visual basic with the use of hide and show. Detailed mockup resembling and behaving close to the final product. 7. Discuss Figure 7. 26 with regard to the user's interaction; the screen's interaction; and the system's interaction (process). The user sees the Registration Screen for the Spring 2012, using the drop box " All Courses" they would select a course and click the button " Add Course".

The screen would show the representation seen at the top of Figure 7. 26 initially. This screen would react to the user's usage of the drop box (showing a scrollable list of he available courses and showing the course selected). When the user has selected a course and clicked " Add Course' the screen would send the information to the registration and when the Registration returns success it would show the second screen in Figure 7. 26. The system's interaction shown in Figure 7. 26 begins with the Registrations object passing the registration information to the section object.

This section object needs to check if the student has the prerequisites for the course. If the Boolean returned is True then the message Add(student) is sent to students object. Note: another sequence diagram would represent Boolean = False. 8. Explain " Adding a course" from section 7. 3. 4. 1 using Normal's 7 Stage Model. 1 Form a goal -? add Engle 201 1 - The user (student) forms the goal " I need to add Engle 2011 to my schedule". 2 Form an intention - find Engle 201 1 The user (student) forms the intention on the screen shown in 7. 26 to find Engle 201 1 in the drop box. Specify the action - mouse can show courses in the drop box - The user will decide to use the mouse to click and hold open the drop box and move the mouse down until he/she sees the Engle 2011 course. Execute the action - The user moves the mouse to the arrow in the drop box and clicks when he/she is on the arrow of the drop box. 5 Perceive the system state (feedback) - he/she sees the drop box open with multiple courses in a list. 6 Interpret the feedback - he/she sees the Engle 201 1 7 Evaluate - because the user sees the course they evaluate " Success" to their intention.

Note: They would continue with the next intention to click " Add Course". 9. List the four components of the GYMS model for user interfaces. The 4 factors of GYMS model are : Goals, Operators, Methods, and Selection Rules 10. Explain how the Model-View-controller architectural style is done using HTML-Script-SQL. The Model-View-Controller (MFC) architectural style is done with HTML as the View of the system and SQL as a database. The database has the model of the system. PH is the script used as the controller for the system. Chapter 8: 1 .

Question: What are the two general characteristics of a design that naturally carry over from requirements? Answer: The design is an evolution from requirements. So the two general characteristics of a) consistency and b) completeness of requirements usually carry over to the design. 2. Question: What is Menace's climatic complexity number a) attempting to measure, and b) what is the climatic complexity number of a program A that contains 3 binary predicates (or branches)? Answer: a) Menace's climatic complexity number measures the control flow complexity of software.) Using the simple formula of (#if binary branches + 1), the climatic number of program A is (3+1) or 4. 3. Question: What is a program slice? Answer: A program slice is the set of statements that can affect the value of some specified variable of interest in the program. 4. Question: What are glue tokens and super glue tokens, and which type nutrients more to cohesion? Answer: Glue tokens are data tokens that lie in more than one data-slice, and super glue tokens are data tokens that lie in every data-slice. The super glue tokens contribute more to cohesion because they lie in every data slice. 5. Question: What is software cohesion?

Answer: Cohesion is a characteristic that describes the degree of relatedness within the software. 6. Question: What are the levels of cohesion? Answer: There are 7 levels of cohesion listed in the order of worst to best: coincidental, logical, temporal, procedural, communicational, sequential, and national. 7. Question: What is software coupling? Answer: Coupling is a characteristic that describes the degree of interdependence or interaction among the parts within the software. 8. Question: What are the levels of coupling? Answer: There are 5 levels of coupling listed in the order of worst to best: content, common, control, stamp, and data. . Question: What are the six design measurements of C-K metrics in 00? Answer: They are a) weighted number of methods per class, b) depth of inheritance tree of the class, c) number of children of a class, d) coupling between objects, e) number of espouses to a class message, and f) lack of cohesion among methods in a class 10. Question: In contrast to general design, what is user interface design interested in? Answer: The general software design is concerned about the software components, their characteristics and their interactions.

User interface design is focused on the interaction between the software and its human users; thus both software and human attributes are of concern in user interface design. 1 1 . Question: List four out of the eight rules of user interface design identified by Seminarian and Pleasant. Answer: There are actually 8: a) consistency, b) roved short cuts, c) informative feedback, d) closure in dialogues, e) simple error handling and prevention, f) permit " re-do", g) support locus of control, h) reduce the amount of short-term memory needs. Any four of the eight would be fine. 12.

Question: What are the general cohesion and coupling goals when designing software? Answer: We strive for lesser or looser coupling of software, and we design for stringer or tighter cohesion. 13. Question: What is the Law of Demeter attempting to achieve? Answer: Law of Demeter is a set of guiding rules that limits the p of control of an object y restricting the messaging structure among methods of a class. Through these restrictions it is conjectured that we can reduce coupling and enhance cohesion. 14. Question: What is fan-in and fan-out, and what attribute do they measure?

Answer: Fan-in is a measure of number of information flow into a module or component. Fan-out is the number of information flow out of a module. An example of fan-in for module-x would be the number of modules that call module-x and for fan-out of module-x would be the number of modules that are called by module-x. Thus fan-in and fan-out measure the coupling attribute among modules. Chapter 9: 1. Question: is the act of transforming the detailed design into a valid program in some programming language. Answer: Implementation. 2. Question: List 3 desirable characteristics of a good implementation.

Answer: Readability; maintainability; performance; traceability; correctness; completeness. 3. Question: True or False? : The most important issues for maintaining a good coding style are: to be consistent and to try to highlight the meaning of your code. Answer: True. 4. Question: You should choose longer names for local variables or entities, but can use shorter ones for global variables or entities. Answer: False. 5. Question: List 3 of the types of code comments described in the book. Answer: Repeat of the code;

explanation of the code; marker in the code; summary of the code; description of the code intent; external references.