

Ryan Somma

Beginning on page 641 in your textbook, please answer the following:

5. Answer the following true/false questions. Explain your answers as necessary.

- *Different action verbs should be used in screen dialogue to describe required keyboard actions in order to add variety and interest.* – **FALSE: consistent language should be used so as not to confuse the user**
- *Most managers are expert users, because they need a high level of PC expertise in order to manage effectively.* – **FALSE: high level manager may have PC expertise, but it is often in a limited set of management-oriented software. John McCain is a high-level manager who does not even know how to use a computer.**
- *Organizations should expect that expert designers, who come highly recommended and who are at the top of their pay scale, will need to refine and modify their user interface designs several times before the result will be satisfactory to the organization.* – **TRUE: prototyping and refactoring a design according to user input and new requirements is part of what makes a good designer**
- *Windows user interface design often borrows from Web interface styles and techniques.* – **FALSE: web designs are more like legacy systems; although, they are emulating Windows more as web design becomes more advanced**
- *Applications need only one type of help menu or dialogue.* – **TRUE if we are talking about a single software application, where one type of help or dialogue is preferable for standardization. FALSE if we are talking about multiple, unrelated applications, where the type of help or dialogue should best conform to the application**
- *Users appreciate clever or humorous screen messages.* – **FALSE a humorous screen message stops being humorous the 100th time you've read it.**
- *The process for designing user interfaces is straightforward and easy to understand.* – **FALSE: charting, state transition diagrams, and prototyping are necessary for designing UIs. These are not intuitive subjects.**

6. In designing user interfaces, consideration must be given to information security and privacy. Describe some of the guidelines and considerations that must be taken into account in building internal controls into the user interface design.

User authentication and authorization is an important dimension to UI. How to assign groups to users, how to ensure they are who they say they are, and deciding what screens and what functions on screens different user groups can perform is an important consideration.

10 At one time, most software applications came with a thick users' manual. Many of these users' manuals have disappeared in favor of sophisticated online help systems and tutorials. If you are a systems designer designing the help system for a new application, what are some of the important considerations to keep in mind?

Make the help documentation immediately accessible from anywhere in the system, and, more importantly, make it context-sensitive—have it go to the chapter or web page concerned with the screen the user is viewing or task they are currently performing. Make the help library searchable and provide pop-up helpers where appropriate so the user doesn't even have to open the help documentation.

Beginning on page 676 in your textbook, please answer the following:

1. What is the main rationale for using object-oriented methods to develop systems. Why?

Object-Oriented methods promote reusability of code. They accomplish this by packaging up functions into classes with methods and properties. These serve as blackbox components that developers can have their code interface with to accomplish specific tasks. This technique prevents developers from having to know anything about the programming logic of a class, and therefore prevent them from having to deal with its complexity, and the objects being able to stand alone promotes loose-coupling, which allows objects to change without impacting the rest of the system.

3. True or false? Explain your answers as needed.

- A dependency relationship models a two-class association in only two instances. – **TRUE: but one class can have multiple dependency relationships**
- To enforce encapsulation, attributes should generally be declared private. –**TRUE: if something in the system needs access to an attribute, the class should provide methods for accessing it.**
- An object that is supposed to collaborate with other objects when necessary to provide a requested service, but which is unable to do so, is termed an irresponsible object. – **FALSE: a Google search of this term finds it nowhere in use in the programming community**
- Interface objects are typically persistent. – **FALSE: persistent objects are stored in a database or other source that preserves their states when the system is not being used, interface objects only provide information for a session.**
- During the object-oriented design phase, the object model is updated to reflect the actual implementation environment. – **TRUE: there are objects that only the system will use that were not apparent in the requirements**

6. Fill in the blanks:

- Window **Controls**, e.g., icons, buttons and links, are **explicitly** stated in system design **use-cases**.
- The term for a set of **related** objects that are **collaborating**, have an **interface** and which can act as a single unit is **component**.
- To be able to **reuse** objects, they need to **couple** correctly by defining them within an appropriate **class** hierarchy so they are **cohesive** enough for easy use in other applications.
- During the **object-oriented design** phase, **class diagram** and **use-case model** are refined to mirror the **implementation** environment of the solution rather than an environment based upon a **modeled** ideal.

7. Match the terms in the first column with the definitions or examples in the second column (see page 677).

1. Visibility	D. External object's access level to an attribute or method
2. Design Pattern	A. Common reusable solution to given problem in given context
3. Components	M. DLL or .exe file
4. State Transition Event	K. Change in state caused by occurrence updating attributes' values
5. Interface object	L. API screen, window, dialogue box
6. Control object	H. Holds business rule or application logic
7. Entity object	J. Representation of business domain's actual data.

8. Object state	C. Object condition at a specific point during its lifetime
9. Object responsibility	E. Obligation to collaborate if needed to provide requested service
10. Object framework.	F. Collaborating objects subsystem providing set of related services.
11. Role playing	I. Acting out use-case scenarios to simulate object behaviors
12. Method	B. Execution of software logic in response to message
13. State machine diagram	G. Model of single object's life cycle states