Week 5 Select Textbook Homework Solutions

Chapter 10

10.1)

The concepts of styles and patterns occur for buildings and software at both macroscopic and microscopic levels. For example, overall styles (center-hall colonial, A-frame) can be found in a house. These represent macroscopic styles. Lower-level microscopic patterns (for a house) can be found in categories of wood molding, fireplace designs, windows, etc

.

10.2)

Data centered architecture: airline reservation system; library catalog system; hotel booking system

Data flow architecture: any engineering/scientific application where computation is the major function

Call and return architecture: any I-P-O application

Object-oriented architectures: GUI-based applications; any OO application

Layered architecture: any application in which the application functions must be decoupled from the underlying OS or network detail. Client server software is often layered.

10.3)

Hierarchical: dataflow, call return, layer

Non-hierarchical: data centered, object-oriented

Non-hierarchical architectures are probably best implemented using object-oriented and event driven programming techniques.

Chapter 11

11.3) The Open-Closed Principle (OCP) states module [component] should be open for extension but closed for modification. The designer creates the designer should specify the component in a way that allows it to be extended (within the functional domain that it addresses) without the need to make internal (code abstractions that serve as a buffer between the functionality that is likely to be extended and the design class itself.

11.4) Dependency Inversion Principle (DIP) states, depend on abstractions. Do not depend on concretions. The more a component depends on other concrete components (rather than on abstractions such as an interface), the more difficult it will be to extend.

11.5)   Within the context of component-level design for object-oriented systems, cohesion implies that a component or class encapsulates only attributes and operations that are closely related to one another and to the class or component itself. Components that highly cohesive are insulated from relying on services provided by other components will make their implementation and maintenance of easier.

11.6)   Coupling is a qualitative measure of the degree to which classes are connected to one another. As classes (and components) become more interdependent, coupling increases. A benefit of low coupling if the ease among components is the ease with which components can be modified without affected the behavior of other components.