Agenda – Module #7
Chapter 7: Structural and Behavioral Design Patterns in Detailed Design

Overview:

As seen in the previous module, common patterns in object-oriented designs exist to provide detailed design solutions to problems that recur many times over in different systems. Beside the creational design patterns studied so far, other common and popular design patterns have been identified to address structural and behavioral problems commonly encountered in software applications. Structural and behavioral designs patterns help identify problems that deal with the structure and behavior of software designs; they prescribe the classes required for their design solution, and interrelationships required to support object creation their behavior. These patterns allow designers to quickly and systematically identify structural layouts of systems (or subsystems) and provide avenues for examining the system's interactions and quality evaluation within the operational system. This module explores several well-established structural and behavioral design patterns and examines the problems they are designed to address, together with the benefits they provide.

Module Objectives:

- *Understand the importance and role of structural and behavioral design patterns.*
- *Identify, understand, and model common structural and behavioral design patterns.*
- *Become proficient in implementing models of both structural and behavioral design patterns.*
- *Understand the benefits of important structural and behavioral design patterns.*

Session 1: Structural Design Patterns - Adapter, Composite, and Facade

This session focuses on structural design patterns in detailed design. The adapter design pattern is presented for allowing systems with incompatible interfaces to work together, therefore increasing the reusability and evolution of software systems. The composite design pattern is presented to allow designers to compose (large) tree-like design structures by strategically structuring objects that share a whole-part relationship, where both whole and part objects are treated uniformly, therefore, operations that are common to both type of objects can be applied the same way to both types of objects. The composite design pattern is applied for the design of a message generator, creating both primitive and composite messages. Finally, the facade design pattern is presented to decrease complexity and coupling among subsystems.

Slides – Chapter 7 – Session I

Session 2: Behavioral Design Patterns – Iterator and Observer

This session focuses on behavioral design patterns in detailed design. The Iterator design pattern is presented for providing a standardized way for accessing and traversing objects in a collection

data structure. By providing a standard interface for accessing elements of a collection structure, client code becomes more consistent and easier to maintain, since changing the internal structure of the data collection structure does not affect the way client code interacts with the structure. This session also presents the observer design patterns, which it is used in many practical applications, where a common design structure is required to support interaction between objects that monitor a common data source so that when changes occur in the data source, the objects react appropriately. The observer design pattern is very popular and prevalent in today's modern languages and frameworks, such as Java (i.e., Observable and Observer interfaces) and .NET.
Slides – Chapter 7 – Session II

Homework #7

- Read chapter 7 and answer review questions 1-11. Submit your answers as a word or PDF document.

Quiz #7
See Quiz #7.