

Agenda – Module #2

Chapter 2: Software Design with the UML

## **Overview:**

Communication is an essential, critical skill for engineers. Throughout a project's life-cycle, software engineers spend a great deal of time and effort communicating with stakeholder, among themselves, and with the computer via programming languages. By unifying the communication language, so that it is appropriate for stakeholder, software engineers, and translation to programming languages, a more efficient design process can be executed and better framing of design problems and their solutions can be achieved. The Unified Modeling Language (UML) provides the mechanisms for creating detailed models that portray a system's design. By providing a visualization method for complex design concepts, communication is enhanced at all phases of the software development life-cycle. This week, we will cover the fundamental concepts of UML to establish a common frame of reference for discussing other important design concepts throughout the course.

### Module Objectives:

- *Understand the role and importance of UML in software design.*
- *Become familiar with UML's common structural and behavioral diagrams.*
- *Understand the relationship between structural UML models and code.*
- *Understand how to model concurrency with UML.*

## **Session 1: UML Fundamentals (30 - 45 minutes)**

This session introduces the fundamental concepts of classifiers, relationships, and enhancing features that are employed in UML diagrams. A brief history of the UML is presented and an overview of the language together with the types of diagrams available for modeling software systems are introduced. The concepts of structural and behavioral modeling are introduced and compelling reasons are presented for studying and employing UML in projects.

Slides – Chapter 2 – Session I

## **Session 2: UML Structural Modeling (30 - 45 minutes)**

This session provides focused discussions on structural modeling of software systems. The concept of quality is briefly introduced as it applies to structural modeling. An important relationship is established between quality and the structure of software systems, so that it becomes evident that quality is affected by structure and structure driven by expected quality. Detailed coverage of component, class, and deployment diagrams is presented as well as examples of forward engineering code from UML diagrams. This session provides the building blocks for future coverage of architecture and detailed design.

Slides – Chapter 2 – Session II

## **Session 3: UML Behavioral Modeling (30 - 45 minutes)**

This session provides focused discussions on behavioral modeling of software systems. The discussion of quality is continued by relating it to performance, which can be evaluated through behavioral modeling. Detailed coverage of use case, communication, and sequence diagrams is presented as well as examples of each. This session provides the building blocks for future coverage of architecture and detailed design and finalizes the discussion on UML.

Slides – Chapter 2 – Session III

### **Homework #2**

- Read chapter 2 and answer review questions 1-19. Submit your answers as a word or PDF document.
- Using the UML (or drawing) tool of choice, complete Chapter Exercises 3 and 4. For both cases, state any assumption that you make during the modeling process. Submit your diagrams together with design assumptions as one single .pdf or word document. Two open-source tools include (that are freely available) are *StarUML* and *ArgoUML*. I strongly recommend that you download and learn one UML tool; however, this is not a requirement. Alternatively, you can use any other drawing product, e.g., MS Word, Visio, etc. to complete the assignment.

### **Quiz #2**

*See Quiz #2.*