**Course Title: Software Design & Architecture Course**

**Code: SE-4343**

**Course Structure: Lectures: 3/Labs: 0**

**Credit Hours: 3**

**Prerequisites: SE-3342 (Software Requirement Engineering)**

**Course Objectives:**

Upon completion of this course, students will develop the ability to apply a wide variety of design patterns, frameworks, and architectures in designing a wide variety of software.

**Course Syllabus:**

Basics of Software Architecture (SA).Characteristics of SA, Importance of SA, SA Business Cycle and Software Processes, Architectural Patterns.SA Case Study. Creating Architecture, Quality Attributes, Business Qualities, Architecture Qualities. Achieving Qualities.Global Analysis.Conceptual Architecture View.Module Architecture View.Execution Architecture View.Code Architecture View.Designing & Documenting the Architecture. Analyzing Architectures.The CBAM.Reconstructing Software Architectures.

**Course Outline:**

1. Software Architecture (SA): SA in Context, SA as a Design Plan, Abstraction, SA Terminology, 4+1 Views Model: Coupling Between Views, Uses and Notation of Four Views. IS2000: The Advanced Imaging Solution, Chapter 1 [TB 1:Ch. 1, Ch. 2]

2. Characteristics of SA, Importance of SA, SA Business Cycle and Software Processes, SA History, "Good" Architecture, Architectural Patterns, Reference Models, and Reference Architectures, Architectural Structures and Views [TB 2: Ch.1, Ch.2]

3. SA Case Study: A-7E Avionics System - Architecture, Business Cycle, Requirements and Qualities [TB 2: Ch. 3]

4. Creating Architecture: Understanding Quality Attributes, Functionality and Architecture, Architecture and Quality Attributes, System Quality Attributes, Quality Attribute Scenarios in Practice, Other System Quality Attributes, Business Qualities, Architecture Qualities [TB 2: Ch. 4]

5. Achieving Qualities: Introducing Tactics, Availability Tactics, Modifiability Tactics, Performance Tactics, Security Tactics, Testability Tactics, Usability Tactics, Relationship of Tactics to Architectural Patterns, Architectural Patterns and Styles [TB 2: Ch. 5]

6. Global Analysis: Overview of Global Analysis Activities, Analyze Factors, Develop Strategies., Analyze Organizational Factors, Begin Developing Strategies, Analyze Technological Factors, Continue Developing Strategies, Analyze Product Factors, Continue Developing Strategies [TB 1: Ch. 3]

7. Conceptual Architecture View: Design Activities for the Conceptual Architecture View, Global Analysis, Central Design Tasks: Components, Connectors, and Configuration, Final Design Task: Resource Budgeting, Traceability, Uses for the Conceptual Architecture View [TB 1: Ch. 4]

8. Module Architecture View: Design Activities for the Module Architecture View, Global Analysis, Central Design Tasks: Modularization and Layering, Final Design Task: Interface Design, Traceability, Uses for the Module Architecture View [TB 1: Ch. 5]

9. Execution Architecture View: Design Activities for the Execution Architecture View, Global Analysis, Central Design Tasks: Runtime Entities, Communication Paths, and Configuration, Final Design Task: Resource Allocation, Traceability, Uses for the Execution Architecture View [TB 1: Ch. 6]

10. Code Architecture View: Design Activities for the Code Architecture View, Global Analysis, Central Design Tasks, Final Design Tasks, Traceability, Uses for the Code Architecture View [TB 1: Ch. 7]

11. Designing & Documenting the Architecture: Architecture in the Life Cycle, Designing the Architecture, Forming Team, Creating a Skeletal System. Documenting Software Architectures, Uses of Architectural Documentation, Views, Choosing the Relevant Views, Documenting a View, Documentation across Views, Unified Modelling Language [TB 2: Ch. 7, Ch. 9]

12. Analyzing Architectures: The ATAM - A Comprehensive Method for Architecture Evaluation, Participants in the ATAM, Outputs of the ATAM, Phases of the ATAM, The Nightingale System: A Case Study in Applying the ATAM [TB 2: Ch. 11]

13. The CBAM: A Quantitative Approach to Architecture Design Decision Making, DecisionMaking Context, The Basis for the CBAM, Implementing the CBAM, Case Study: The NASA ECS Project, Results of the CBAM Exercise [TB 2: Ch. 12]

14. Reconstructing Software Architectures: Introduction, Information Extraction, Database Construction, View Fusion, Reconstruction [TB 2: Ch. 10]

**Class Assignments:**

* Planning and Practice of existing software design methodologies.
* Outline of requirements, the existing design & architecture practices using up to date tools and technologies.
* Applications of Software Development Life Cycle (SDLC), its phases and thus implementation of different process models.
* Efficient use of different modeling and design tools e.g. UML (for code generation), open source code development etc.
* Writing of Software Design Specifications.

**Textbook(s):**

1. Applied Software Architecture by Christine Hofmeister, Robert Nord and DilipSoni, Addison-Wesley Professional (1999). ISBN-10: 0201325713.
2. Software Architecture in Practice by Len Bass, Paul Clements and Rick Kazman, Addison-Wesley Professional; 2ndEdition (April 19, 2003). ISBN-10: 0321154959

**Reference Material:**

* Software Architecture in Practice by Len Bass, Paul Clements and Rick Kazman, Addison-Wesley Professional; 3rdEdition (2012). ISBN-10: 0321815734
* Software Architecture and Design Illuminated by Kai Qian, Xiang Fu, Lixin Tao and Chong-weiXu, Jones & Bartlett Publishers; 1stEdition (2009). ISBN-10: 076375420X
* Software Architecture: Foundations, Theory, and Practice by R. N. Taylor, N. Medvidovic and E. M. Dashofy, Wiley; 1stEdition (2009). ISBN-10: 0470167742.