**Course Name: Software Design and Architecture**

**Credit Hours**: 3

**Prerequisites**: **Software Engineering**

**Objectives:** An in-depth look at software design. Continuation of the study of design patterns, frameworks, and architectures. Survey of current middleware architectures. Design of distributed systems using middleware. Component based design. Measurement theory and appropriate use of metrics in design. Designing for qualities such as performance, safety, security, reusability, reliability, etc. Measuring internal qualities and complexity of software. Evaluation and evolution of designs. Basics of software evolution, reengineering, and reverse engineering.

Upon completion of this course, students will have the ability to:

* Apply a wide variety of design patterns, frameworks, and architectures in designing a wide variety of software
* Design and implement software using several different middleware technologies
* Use sound quality metrics as objectives for designs, and then measure and assess designs to ensure the objectives have been met
* Modify designs using sound change control approaches

Use reverse engineering techniques to recapture the design of software

**Outcomes:**

1. Argue the importance and role of software architecture in large-scale software systems.
2. Design and motivate software architecture for large-scale software systems.
3. Recognize major software architectural styles, design patterns, and frameworks.
4. Describe a software architecture using various documentation approaches and architectural description languages.
5. Generate architectural alternatives for a problem and selection among them.
6. Use well-understood paradigms for designing new systems.
7. Identify and assess the quality attributes of a system at the architectural level.
8. Motivate the architectural concerns for designing and evaluating a system's architecture.
9. Discuss and evaluate the current trends and technologies such as model-driven and service-oriented architectures.
10. Evaluate the coming attractions in software architecture research and practice.

**Course Outline:**

Introduction to the discipline of design, generic design processes, and design management; software product design, including analysis activities such as needs elicitation and documentation, requirements development activities such as requirements specification and validation, prototyping, and use case modelling; engineering design analysis, including conceptual modelling and both architectural and detailed design; survey of patterns in software design, including architectural styles and common mid-level design patterns.

**Reference Materials:**

1. *Software Architecture and Design Illuminated*, Kai Qian, Xiang Fu, Lixin

Tao, Chong-Wei Xu, Jorge L. Diaz-Herrera, Jones and Bartlett Publishers,

1st Edition, 2009 (or Latest Edition).

2. *Introduction to Software Engineering Design: Processes, Principles and*

*Patterns with UML2*, Christopher Fox, Addison-Wesley Professional, 2006

(or Latest Edition).

3. *Software Engineering Design: Theory and Practice*, Carlos Otero, CRC

Press, 2012 (or Latest Edition).

4. *Software Engineering Techniques: Design for Quality*, Krzysztof Sacha,

Springer, 2006 (or Latest Edition).