

Software Design and Architecture

Code SE- 211	Credit Hours 2-1
------------------------	----------------------------

Course Description

The Software Design & Architecture course is designed to help students to develop skills that will enable them to construct software of high quality, software that is reliable, easy to understand, modify and maintain. The course is important in relation to the core modules and helps the students to understand the Object-oriented software design and to develop reusable software.

The course introduces principles of good design, object-oriented software design and a number of architectural styles like enterprise architecture, business architecture, Information System Architecture, and technology architecture. These architectural patterns help the students to analyze the non-functional requirements. A particular emphasis is laid on the design process so that students can recognize the significance of design in maintaining the project.

To enhance the abilities of students to develop reusable software designs. To introduce the students to the principles of good design, design approaches, paradigms and object-oriented concepts. In addition, students will learn software architecture that represents the gross-level structure of software intensive systems and includes early design decisions that impact the quality of the overall system. Enterprise architecture is generally considered to play a fundamental role in coping with the inherent difficulties of the development of large-scale and complex software systems. A common assumption is that architecture design can support the required software system qualities such as robustness, adaptability, reusability and maintainability. This course enables the students to apply architecture pattern, design principles on different real-world problems.

Text Book:

1. Java Design Patterns: A Hands-on Experience with Real-World Examples, second edition, 2018, Vaskaran Sarcar
2. Humberto Cervantes, Rick Kazman. Designing Software Architectures: A Practical Approach 2016 Addison Wesley

Reference Book:

1. Alexander Shvets Design Patterns Explained Simply

Prerequisites

SE-205

ASSESSMENT SYSTEM FOR THEORY

Quizzes	10%
Assignments	5-10%
Mid Terms	35%
ESE	50%

ASSESSMENT SYSTEM FOR LAB

Lab Work and Report	70-80%
Lab ESE/Viva	20-30%

Teaching Plan

Week No	Topics	Learning Outcomes
Week1	Introduction to Software Design & Architecture <ul style="list-style-type: none"> • What's a Design Pattern • What does the pattern consist of? • Importance of Design Patterns • Levels of Designs • Design Approaches • Design Challenges • Characteristics of Good and Bad design 	Understand the fundamental concepts of software design and architecture. Recognize and define what design patterns are and their purpose. Explain the significance and benefits of using design patterns in software development. Differentiate between various levels of design in software engineering. Compare different approaches to software design. Identify common challenges faced during software design. Assess the characteristics that define good and bad software design.
Week 2	UML Revision <ul style="list-style-type: none"> • Invariants, Pre- and post conditions OOP Revision	Refresh knowledge on UML and its applications in software design. Understand and apply invariants, pre-conditions, and post-conditions in software specifications.
Week 3	Symptoms of rotten Software Design	Identify and diagnose common symptoms of poor software design.
Week 4	Software Design Process and Principles I Class Level Software Design Principles (SOLID)	Understand the foundational principles and processes involved in software design. Learn and apply SOLID principles to ensure robust and maintainable class-level designs.
Week 5	Software Design Process and Principles II	Delve deeper into advanced software design processes and principles.
Week 6	Software Design Process and Principles II <ul style="list-style-type: none"> • Cohesion and Coupling • Different types of Cohesion and Coupling • Package Level Software Design Principles 	Delve deeper into advanced software design processes and principles. Define and differentiate between cohesion and coupling in software design. Identify and explain various types of cohesion and coupling. Apply design principles at the package level to ensure modular and scalable software systems.

Week 7	Creational Design Patterns I	Understand and implement basic creational design patterns.
Week 8	Creational Design Patterns II	Master more advanced creational design patterns and their applications.
Week 9	Mid Semester Exam	Assess understanding and application of concepts covered in the first half of the course.
Week 10	Structural Design Patterns I	Learn and apply fundamental structural design patterns.
Week 11	Structural Design Patterns II	Explore advanced structural design patterns and their real-world applications.
Week 12	Behavioural Design Patterns I	Understand and implement basic behavioural design patterns.
Week 13	Behavioural Design Patterns II	Master more advanced behavioural design patterns and their applications.
Week 14	Architectural patterns and Styles	Identify and compare different architectural patterns and styles used in software engineering.
Week 15	Enterprise Architecture (EA) Overview	Gain a comprehensive understanding of enterprise architecture concepts and frameworks.
Week 16	Business Architecture Overview, Information System Architecture Overview	Understand the key components and structure of business and information system architectures.
Week 17	Projects	Apply learned concepts and principles to practical projects, demonstrating the ability to design and implement robust software systems.
Week 18	End Term Exam	Evaluate overall understanding and application of course material.

Practical:

Experiment No	Description
01	UML Revision: Use Case Diagrams
02	OOP Revision: Pillars of OOP
03	OO Design Principles (Class and Package level)
04	OO Design Principles (Class and Package level)
05	Creational Design Patterns I
06	Creational Design Patterns II
07	Structural Design Patterns I
08	Lab Assessment
09	Structural Design Patterns II
10	Behavioral Design Patterns I
11	Behavioral Design Patterns II
12	ARIS Workspace Walk-through & Familiarization
13	Defining Simple Business Architecture In ARIS
14	Defining Application Architecture In ARIS
15	Defining Technology Architecture in ARIS
16	Lab Discussion