



# INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal - 500 043, Hyderabad, Telangana

## COURSE CONTENT

SOFTWARE ARCHITECTURE AND DESIGN PATTERNS								
VI Semester: CSE / IT								
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P		CIA	SEE	Total
ACSE29	Elective	3	0	0	3	40	60	100
<b>Contact Classes: 48</b>	<b>Tutorial Classes: Nil</b>	<b>Practical Classes: Nil</b>			<b>Total Classes: 48</b>			
<b>Prerequisite: Object Oriented Programming</b>								

### I. COURSE OVERVIEW:

This course focuses on the principles, concepts, and techniques used in designing and architecting software systems. It emphasizes understanding the critical decisions and trade-offs involved in designing robust, scalable, maintainable, and high-performance software architectures. Students will explore various architectural patterns, design principles, and best practices, along with real-world case studies to understand how architectural decisions impact software development and lifecycle.

### II. COURSE OBJECTIVES:

#### The students will try to learn:

- I To introduce the fundamental concepts and importance of software architecture in the software development lifecycle.
- II To understand architectural styles and design principles that guide the structure and behavior of software systems.
- III To explore the role of architectural documentation, evaluation, and quality attributes in designing scalable and maintainable software.

### III. COURSE OUTCOMES:

#### At the end of the course students should be able to:

- CO1 Understand the principles and significance of software architecture in the software development lifecycle
- CO2 Analyze various architectural styles and patterns for structuring complex software systems.
- CO3 Apply design principles and architectural documentation techniques for scalable software design.
- CO4 Identify and use appropriate design patterns (creational, structural, behavioral) to solve recurring software problems.
- CO5 Evaluate software architecture decisions based on quality attributes such as performance, modifiability, and usability.
- CO6 Design and document software architectures and component-based systems using suitable tools and patterns.

### IV. COURSE CONTENT:

#### MODULE - I: INTRODUCTION TO SOFTWARE ARCHITECTURE

Overview of software architecture definition and importance role of a software architect architectural patterns vs. design patterns, architectural styles layered architecture, client-server architecture, event-driven architecture, microservices architecture case studies in software architecture real-world examples analysis and discussion.

#### MODULE - II: ARCHITECTURAL PATTERNS

Introduction to architectural patterns, definition and purpose overview of common patterns, layered pattern, concept and structure, use cases and examples, client-server and microservices patterns, detailed study of each pattern advantages and disadvantages, event-driven and service-oriented architectures, key concepts and implementation examples and best practices.

#### MODULE - III: DESIGN PRINCIPLES AND PRACTICES

Solid principles, single responsibility principle, open/closed principle IISKOV substitution principle interface segregation principle, dependency inversion principle, grasp principles general responsibility assignment software

patterns, information expert, creator, controller low coupling, high cohesion, design documentation diagrams design documentation best practices.

#### **MODULE – IV: DESIGN PATTERNS**

Introduction to Design Patterns, Definition and importance Types of design patterns (Creational, Structural, Behavioral) Creational Patterns, Singleton, Factory Method, Abstract Factory, Builder, Prototype Structural Patterns, Adapter, Composite, Proxy, Flyweight, Facade, Bridge, Decorator Behavioral Patterns, Strategy, Observer, Command, Mediator, Chain of Responsibility, Template Method, State

#### **MODULE – V: APPLYING ARCHITECTURE AND DESIGN PATTERNS**

Pattern Application and Anti-Patterns, identifying suitable patterns for problems, recognizing and avoiding anti-patterns, Refactoring and Design Improvements, Code smells, Techniques for refactoring, Capstone Project Design, and implementing a small software system using the principles and patterns learned.

#### **V. TEXT BOOKS:**

1. Mary Shaw, David Garlan, *Software Architecture: Perspectives on an Emerging Discipline*, Pearson, 1st Edition, 1996
2. Len Bass, Paul Clements, Rick Kazman, *Software Architecture in Practice*, Addison-Wesley, 4th Edition, 2021
3. Eric Gamma, Richard Helm, Ralph Johnson, John Vlissides (Gang of Four), *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1st Edition

#### **VI. REFERENCE BOOKS:**

1. Clemente Minossi, *Software Architecture with Python*, Packt Publishing, 1st Edition, 2021
2. Mahesh P. Matha, *Software Architecture and Design Patterns*, PHI Learning, 1st Edition, 2008

#### **VII. ELECTRONICS RESOURCES:**

1. <https://c4model.com/>
2. <https://www.coursera.org/learn/software-design-architecture>
3. <https://www.coursera.org/specializations/cloud-architecture>

#### **VIII. MATERIAL ONLINE:**

1. Course Outline Description
2. Lecture notes
3. PowerPoint presentation
4. Definitions and Terminology
5. Tutorial Question Bank
6. Case Studies
7. Real life Examples
8. Complex Engineering Problems
9. Tech Talk Topics
10. Concept Video Topics
11. Open-ended Exercises
12. Assignments
13. Model Question Paper – I
14. Model Question Paper – II
15. GATE Question Bank
16. Previous Question Papers and Solutions