

Subject Code	Subject Name	Credits
26CS502	SOFTWARE ARCHITECTURE AND ADVANCED SOFTWARE ENGINEERING	4

Course Objectives:

1. To understand software architecture principles and design patterns
2. To design scalable, maintainable, and high-performance systems
3. To study modern architectural styles (microservices, cloud-native, etc.)
4. To explore quality attributes like reliability, security, and scalability
5. To gain knowledge of DevOps, CI/CD, and software lifecycle management

Learning Objectives:

After completion of the course the students will be able to:

1. Analyse and design complex software architectures
2. Apply architectural patterns and design principles
3. Evaluate systems based on performance, scalability, and security
4. Implement microservices and distributed systems
5. Integrate DevOps practices in software development
6. Manage large-scale software projects effectively

Unit I - Introduction to Software Architecture (12 Hrs.)

Definition of software architecture, Role of architecture in software systems, Evolution of software architecture, Importance in large-scale systems, Architecture as a blueprint, Components and connectors and importance of software architecture, Architecture vs design, Stakeholders and concerns, Architectural views and viewpoint

Unit II - Architectural Patterns & Styles (12 Hrs.)

Layered architecture, Client-server architecture, Microservices architecture, Event-driven architecture, Service-Oriented Architecture (SOA), Design Principles & Patterns, SOLID principles, Object-oriented design principles, Design patterns (Creational, Structural, Behavioral), Dependency injection and inversion of control

Unit III - Advanced Software Engineering Practices (12 Hrs.)

Agile methodologies (Scrum, Kanban), DevOps concepts, Continuous Integration / Continuous Deployment (CI/CD), Version control systems (Git) Software Quality & Architecture Evaluation: Quality attributes (performance, reliability, security, usability), Architecture evaluation methods (ATAM), Technical debt, Maintainability and testing strategies

UNIT IV - Security & Performance Engineering (12 Hrs.)

Secure software design principles - Introduction to software security, Security design principles: Secure coding practices, Authentication and authorization, Data encryption (at rest & in transit) Secure API design, Threat modeling, Performance optimization techniques, Load balancing and caching

UNIT V - Emerging Trends in Software Architecture (12 Hrs.)

Definition of serverless computing - Function as a Service (FaaS), Event-driven execution model, Limitations of serverless, Use cases (APIs, data processing) Serverless architecture, Edge computing, AI in software engineering, Blockchain-based architectures

Reference Books:

1. Dusty Phillips, Python 3 Object-Oriented Programming, Packet Publishing.
2. Steven F. Lott, Mastering Object-Oriented Python, Packet Publishing.
3. Luciano Ramalho, Fluent Python, O'Reilly Media.
4. Mark Lutz, Learning Python, O'Reilly Media.
5. Allen B. Downey, Think Python: How to Think Like a Computer Scientist, O'Reilly Media.