

Uka Tarsadia University



B. Tech.
Semester V

**SOFTWARE ARCHITECTURAL STYLES AND
DESING PATTERNS
CE5019**

EFFECTIVE FROM June-2023

Syllabus version: 1.00

Subject Code	Subject Title	Teaching Scheme			
		Hours		Credits	
		Theory	Practical	Theory	Practical
CE5019	Software Architectural Styles and Design Patterns	3	2	3	1

Subject Code	Subject Title	Theory Examination Marks		Practical Examination Marks	Total Marks
		Internal	External	CIE	
CE5019	Software Architectural Styles and Design Patterns	40	60	50	150

Objectives of the course:

- To expose students with the concepts of software architectures and its design.
- To describe design patterns and its significance.

Course outcomes:

Upon completion of the course, the student shall be able to,

CO1: Explain the significance of software architectures.

CO2: Understand and apply quality attributes of software architectures.

CO3: Understand and apply agile practices in architecture.

CO4: Explain design patterns and its significance.

CO5: Understand and apply creational and structural patterns.

CO6: Understand and apply behavioural patterns.

Sr. No.	Topics	Hours
Unit – I		
1	Software Architectures: Introduction to software architectures, Architectural structures and view, Architectural patterns, Characteristics of good software architectures, Importance of software architecture, Architecture in - Technical context, Project life-cycle context, Business context, and Professional context.	5
Unit – II		

2	Quality Attributes of Architecture: Architecture and Requirements, Functionality, Consideration of quality attributes, Specifying requirements of quality attributes, Availability, Interoperability, Modifiability, Performance, Security, Testability, Usability, Other quality attributes.	6
Unit – III		
3	Architecture in Agile Projects: Agility and Architecture methods, Guidelines of Agile architecture, Gathering architectural requirements from – Requirements document, Interviewing stockholders, Understanding the business goals and Utility tree, Designing and Documenting architecture, Architecture implementation and testing.	7
Unit – IV		
4	Design Patterns: Introduction to design patterns, Design patterns in smalltalk MVC, Describing design patterns, The catalog of design patterns, Organizing the design pattern catalog, Problem solving with design patterns, Selection of design patterns, Use of design patterns, Case study.	5
Unit – V		
5	Creational patterns: Abstract factory, Builder, Factory method, Prototype, Singleton, Case study. Structural patterns: Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Proxy, Case study.	7
Unit – VI		
6	Behavioural patterns: Chain of responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, template method, Visitor, Case study.	6

Text books:

1. Len Bass, Paul Clements, Rick Kazman, "Software Architecture in Practice", Addison-Wesley.
2. Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, "Design Patterns", Addison Wesley.

Reference books:

1. Bernd Bruegge, Allen H. Dutoit, "Object-oriented Software Engineering Using UML, Patterns, and Java", Pearson.

2. Kamon Ayeva and Sakis Kasampalis, "Mastering Python Design Pattern", Packt Publication.
3. Mark Richards, "Software Architecture Patterns", O'Reilly.
4. Martin Fowler and Scott, "UML Distilled", Addison-Wesley.

Course objectives and Course outcomes mapping:

- To expose students with the concepts of software architectures and it's design: CO1, CO2, and CO3.
- To describe design patterns and its significance: CO3, CO4, and CO5.

Course units and Course outcomes mapping:

Unit No.	Unit Name	Course Outcomes					
		CO1	CO2	CO3	CO4	CO5	CO6
1	Software Architectures	✓					
2	Quality Attributes of Architecture		✓				
3	Architecture in Agile Projects			✓			
4	Design Patterns				✓		
5	Creational and structural patterns					✓	
6	Behavioural patterns						✓

Programme outcomes:

- PO 1: Engineering knowledge: An ability to apply knowledge of mathematics, science, and engineering.
- PO 2: Problem analysis: An ability to identify, formulates, and solves engineering problems.
- PO 3: Design/development of solutions: An ability to design a system, component, or process to meet desired needs within realistic constraints.
- PO 4: Conduct investigations of complex problems: An ability to use the techniques, skills, and modern engineering tools necessary for solving engineering problems.
- PO 5: Modern tool usage: The broad education and understanding of new engineering techniques necessary to solve engineering problems.
- PO 6: The engineer and society: Achieve professional success with an understanding and appreciation of ethical behavior, social responsibility, and diversity, both as individuals and in team environments.

- PO 7: Environment and sustainability: Articulate a comprehensive world view that integrates diverse approaches to sustainability.
- PO 8: Ethics: Identify and demonstrate knowledge of ethical values in non-classroom activities, such as service learning, internships, and field work.
- PO 9: Individual and team work: An ability to function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- PO 10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give/receive clear instructions.
- PO 11: Project management and finance: An ability to demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- PO 12: Life-long learning: recognition of the need for, and an ability to engage in life-long learning.

Programme outcomes and Course outcomes mapping:

Programme Outcomes	Course Outcomes					
	C01	C02	C03	C04	C05	C06
P01						
P02						
P03		✓	✓			
P04						
P05				✓	✓	✓
P06						
P07						
P08						
P09						
P010						
P011		✓		✓	✓	✓
P012						