IV B.Tech – I Semester
(17CS702) SOFTWARE ARCHITECTURE & DESIGN PATTERNS

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
40	60	100	3	1	-	3

Pre-Requisites: Fundamentals of Software Architecture

Course Objectives:
The course should enable the student:
- To understand interrelationships, principles and guidelines governing architecture and evolution over time.
- To understand various architectural styles of software systems.
- To understand design patterns and their underlying object oriented concepts.
- To understand implementation of design patterns and providing solutions to real world software design problems.
- To understand patterns with each other and understanding the consequences of combining patterns on the overall quality of a system.

UNIT-I: Envisioning Architecture the Architecture Business Cycle, what is Software Architecture, Architectural patterns, reference models, reference architectures, architectural structures and views. Creating and Architecture Quality Attributes, achieving qualities, Architectural styles and patterns, designing the Architecture, Documenting software architectures, Reconstructing Software Architecture.

UNIT-II: Analysing Architectures Architecture Evaluation, Architecture design decision making, ATAM, CBAM Moving from One System to Many Software Product Lines, Building systems from off the shelf components, Software architecture in future.

UNIT-III: Patterns Pattern Description, organizing catalogues, role in solving design problems, Selection and usage. Creational Patterns Abstract factory, Builder, Factory method, Prototype, Singleton

UNIT-IV: Structural Patterns Adapter, Bridge, Composite, Decorator, Façade, Flyweight, PROXY.

UNIT-V: Behavioural Patterns, Chain of responsibility, command, Interpreter, iterator, mediator, memento, observer, state, strategy, template method, visitor.

UNIT-VI: Case Studies A-7E – A case study in utilizing architectural structures, The World Wide Web - a case study in Interoperability, Air Traffic Control – a case study in designing for high availability, Celsius Tech – a case study in product line development. A Case Study (Designing a Document Editor): Design Problems, Document Structure, Formatting, Embellishing the User Interface, Supporting Multiple Look-and-Feel Standards, Supporting Multiple Window Systems, User Operations, Spelling Checking and Hyphenation.

Course Outcomes:

CO-1	Understand the architecture, creating it and moving from one to any, different structural patterns.	L2
CO-2	Analyse the architecture and build the system from the components.	L4
CO-3	Learns that design patterns are solutions, and they can solve many problems that can be encountered in the future.	L3
CO-4	Understands the structure of design patterns	L3
CO-5	Gains Knowledge on behavioural design patterns	L3
CO-6	Do a case study in utilizing architectural structures and design patterns	L3

Correlation of COs with POs & PSOs:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3
CO-1	-	-	3	-	2	-	-	-	1	-	1	-	-	2	-
CO-2	-	-	3	-	2	-	-	-	1	-	1	-	-	2	-
CO-3	-	-	3	-	2	-	-	-	1	-	1	-	-	2	-
CO-4	-	-	3	-	2	-	-	-	1	-	1	-	-	2	-
CO-5	-	-	3	-	2	-	-	-	1	-	1	-	-	2	-
CO-6	-	-	3	-	2	-	-	-	1	-	1	-	-	2	-

Text Books:

1. Software Architecture in Practice, second edition, Len Bass, Paul Clements & Rick Kazman, Pearson Education, 2003.
2. Design Patterns, Erich Gamma, Pearson Education, 1995.

Reference Books:

1. Beyond Software architecture, Luke Hohmann, Addison wesley, 2003.
2. Software architecture, David M. Dikel, David Kane and James R. Wilson, Prentice Hall PTR, 2001
3. Software Design, David Budgen, second edition, Pearson education, 2003
4. Head First Design patterns, Eric Freeman & Elisabeth Freeman, O'REILLY, 2007.
5. Design Patterns in Java, Steven John Metsker & William C. Wake, Pearson education, 2006
6. J2EE Patterns, Deepak Alur, John Crupi & Dan Malks, Pearson education, 2003.
7. Design Patterns in C#, Steven John metsker, Pearson education, 2004.
8. Pattern Oriented Software Architecture, F.Buschmann & others, John Wiley & Sons.