Course code	Course Name	L-T-P-Credits	Year of Introduction
IT301	**Software Architecture and Design Patterns**	3-1-0 - 4	2016

Prerequisite : Nil

Course Objectives
- To introduce to the students the basic knowledge of software, software development process and the concepts of software design principles.
- Gain knowledge on how to design UML diagrams.
- To impart knowledge on the different architectural styles and architectural patterns for the software.

Syllabus

Introduction to the concepts of Software and the software design process, Process models, Importance of software architecture, Software design principles, Introduction to UML diagrams, UML diagram preparation for various case studies. Introduction to Software Architectural styles, Software Architecture patterns, Evaluation of architectural design.

Expected outcome

The students will be able to

i.   Design UML diagram for the software.

ii.   Identify and apply appropriate architectural styles and architectural design pattern for the software.

iii. Create flexible, reusable and efficient architecture for software.

Reference Books:
1. Eric J. Braude , Software Design, John Wiley and Sons.
2. Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides : Design Patterns: Elements of Reusable Object-Oriented Software, Addison – Wesley, 1994
3. James Rumbaugh, Object Oriented Modeling and Design, Prentice Hall India
4. Len Bass, Paul Clements, Rick Kazman, Software Architecture in Practice (2nd Ed.), Pearson
5. Mary Shaw & David Garlan, Software Architecture – Perspectives on an emerging discipline, Pearson, 1996
6. Roger S.Pressman, Software Engineering – A Practitioner's approach, 8th edition(2014), McGraw Hill Education

Course Plan			
Module	**Contents**	**Hours**	**Sem. Exam Marks**
I	The Nature of Software – Defining software, Software Application domains. Software Engineering. The Software Process, Process Models: - Water fall model, Incremental model, Prototyping model, RAD, Spiral model and Agile Development. (Brief introduction of Agile Development). What is Software Architecture? Why is software architecture important? Role of software architect.	8	15%

II	Software Design principles – Correctness and Robustness – Flexibility, Reusability and Efficiency – Tradeoffs among robustness, flexibility, reusability and efficiency	7	15%
FIRST INTERNAL EXAM			
III	Introduction to UML diagrams – Use case diagrams, Class diagrams, Sequence diagrams, Activity diagrams, State Transition diagram, Deployment diagram.	8	15%
	Case study discussion on UML diagrams – Group presentation by students on different case studies.	4	
IV	Software architectural styles - pipes & filters, layered, event-based, data-centered, interpreter, MVC, message dispatcher, multi-tier distributed	9	15%
SECOND INTERNAL EXAM			
V	What is a design pattern? Creational patterns – Factory, Abstract Factory, Prototype and Singleton. Structural patterns – Composite, Decorator, Adapter, Façade and Flyweight.	9	20%
VI	Behavioral patterns- Chain of responsibility, Command, Interpreter, Mediator, State, Template and Observer. Evaluation of architectural design - ATAM	9	20%
END SEMESTER EXAM			

QUESTION PAPER PATTERN

Maximum Marks: 100   Exam Duration: 3 hours

The question paper shall consist of Part A, Part B and Part C.

Part A shall consist of three questions of 15 marks each uniformly covering Modules I and II. The student has to answer any two questions (15×2=30 marks).

Part B shall consist of three questions of 15 marks each uniformly covering Modules III and IV. The student has to answer any two questions (15×2=30 marks).

Part C shall consist of three questions of 20 marks each uniformly covering Modules V and VI. The student has to answer any two questions (20×2=40 marks).

Note : Each question can have a maximum of 4 subparts, if needed