SOFTWARE ARCHITECTURE AND DESIGN PATTERNS
SYLLABUS

Offering Branches	IT	**Course Code**	20IT6601
Course Category	Honors	**Credits**	4
Course Type	Theory	**Lecture-Tutorial-Practical**	4-0-0
Prerequisites	Software Engineering	**Continuous Evaluation**	30
		Semester End Evaluation	70
		Total Marks	100

Course Outcomes		
Upon successful completion of the course, the student will be able to		
CO1	Demonstrate an understanding of a range of design patterns.	**L2**
CO2	Experience core design principles and to assess the quality of a design with respect to these principles.	**L3**
CO3	Capable of applying these principles in the design of object oriented systems.	**L3**
CO4	Select and apply suitable patterns in specific contexts	**L3**

Syllabus		
Unit No	**Contents**	**Mapped CO**
I	**Introduction:** what is a design pattern? Describing design patterns, the catalog of design pattern, organizing the catalog, how design patterns solve design problems, how to select a design pattern, how to use a design pattern.	**CO1**
II	**Analysing a System:** overview of the analysis phase, stage 1: gathering the requirements- functional requirements specification, defining conceptual classes and relationships, Design (Library system).	**CO2, CO3**
III	**Structural patterns**: Adapter, bridge, composite, decorator, façade.	**CO1- CO4**
IV	**Behavioral Patterns:** Chain of Responsibility, Command, Interpreter, Iterator, Template Method	**CO1- CO4**
V	**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	**CO1- CO4**

Learning Resources
Text Books
1. Brahma Dathan, Sarnath Rammath, Object-oriented analysis, design and implementation, UniversitiesPress, 2013 2. Erich Gamma, Richard Helan, Ralph Johman, John Vlissides , Design Patterns, Pearson Publication, 2013
References
1. Frank Bachmann, Regine Meunier, Hans Rohnert “Pattern Oriented Software Architecture” – Volume 1, 1996. 2. William J Brown et al., "Anti-Patterns: Refactoring Software, Architectures and Projects in Crisis", JohnWiley, 1998.

Course Coordinator **HOD**