

<b>Software Modeling and Analysis</b>	Code & No:	CS 432
	Credits:	<u>3 (3,0,1)</u>
	Pre-requisite:	CS 360
	Co-requisite:	None
	Level:	10

**Course Description:**

This course presents an integrated set of techniques for software analysis and design based on object-oriented concepts and the UML notation. Topics include introduction to object concepts, fundamentals of object oriented analysis and design process, use-case analysis, object modeling using behavioral techniques, design patterns, design quality and metrics

**Course Aims:**

- 1) To introduce the fundamentals of structural analysis and structural design, and object-oriented analysis and design
- 2) To provide software analysis and modeling experience using CASE tools.
- 3) Develop specific skills, competencies, and points of view needed by professionals in the field most closely related to software analysis and design

**Student Outcomes (SOs):**

- (a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline
- (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
- (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
- (d) An ability to function effectively on teams to accomplish a common goal
- (e) An understanding of professional, ethical, legal, security and social issues and responsibilities
- (f) An ability to communicate effectively with a range of audiences
- (g) An ability to analyze the local and global impact of computing on individuals, organizations, and society
- (h) Recognition of the need for and an ability to engage in continuing professional development
- (i) An ability to use current techniques, skills, and tools necessary for computing practice.

(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices. [CS]

(k) An ability to apply design and development principles in the construction of software systems of varying complexity. [CS]

(j) An ability to use and apply current technical concepts and practices in the core information technologies of human computer interaction, information management, programming, networking, and web systems and technologies. [IT]

(k) An ability to identify and analyze user needs and take them into account in the selection, creation, evaluation, and administration of computer-based systems. [IT]

(l) An ability to effectively integrate IT-based solutions into the user environment. [IT]

(m) An understanding of best practices and standards and their application. [IT]

(n) An ability to assist in the creation of an effective project plan. [IT]

**Course Learning Outcomes (CLOs):**

1. Understand the role of analysis and design in the software engineering lifecycle
2. Develop object-oriented designs by applying established design principles
3. Develop use-case and scenario descriptions of the requirements
4. Develop richer descriptions of design models using UML diagrams
5. Understand the role and influence of design patterns and frameworks in software design

**SOs and CLOs Mapping:**

CLO/SO	a	b	c	d	e	f	g	h	i	j	k	l	m	n
CLO1		√												
CLO2			√											
CLO3									√					
CLO4									√					
CLO5		√												

No.	Topics	Weeks	Teaching hours
-----	--------	-------	----------------

1	<u>Introduction to Object –oriented Analysis and Design</u>	1	3
2	<u>Iterative, Evolutionary, and Agile Models</u>	2	6
3	<u>UML Notation</u>	1	3
4	<u>Use Case Diagrams, System Sequence Diagrams, Interaction diagrams</u>	2	6
5	<u>Class Diagrams, Designing Objects with Responsibilities</u>	2	6
6	<u>Mapping Designs to Code, Test-Driven development and Refactoring</u>	2	3
7	<u>GoF Design Patterns</u>	2	6
8	<u>Activity Diagrams, State Diagrams</u>	2	6
<b>Total</b>		<b>14</b>	<b>42</b>

**Textbook:**

- Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development”, by Larman, Craig, 3rd edition, 2008

**Essential references:**

- Design Patterns: Elements of reusable object-oriented software by Gamma, Helm, Johnson and Vlissides. Addison Wesley
- UML Distilled: A brief guide to the standard object modeling language”, by Fowler, Martin 3rd edition, 2004