

Software Architectures

Code & No: CS 435

Credits: 3(3,0,1)

Pre-requisite: [CS 360](#)

Co-requisite: None

Level: 9 or 10

Course Description:

This course introduces basic concepts and principles about software architecture. It starts with an overview of architectural structures and styles. Practical approaches and methods for creating and analyzing software architecture are presented. The emphasis is on the interaction between quality attributes and software architecture. It includes:

- [Software architecture and its importance](#)
- [Contexts of software architectures](#)
- [Understanding Quality attributes of software architecture](#)
- [Architectural tactics and patterns](#)
- [Architecture in the life cycle-Architecture and requirements-Designing an architecture](#)
- [Documenting software architectures](#)
- [Architecture implementation and testing](#)
- [Architecture evaluation](#)
- [Architecture in the cloud](#)

Course Aims:

- 1) Understanding of the concept of software architecture and how this phase in the development between requirement specification and detailed design plays a central role for the success of a software system.
- 2) Knowledge of some well-known architecture patterns.
- 3) Some understanding how the developers' experiences and the technical and organizational environment will influence on the choice of architecture.

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]
- (l) 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]
- (m) 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]
- (n) An ability to effectively integrate IT-based solutions into the user environment. [IT]
- (o) An understanding of best practices and standards and their application. [IT]
- (p) An ability to assist in the creation of an effective project plan. [IT]

Course Learning Outcomes (CLOs):

1. Able to apply the knowledge of software architectures in building applications
2. Able to distinguish between various architectural styles
3. Able to design, construct and evaluate architectures for software systems
4. Understand the role of architecture in the life cycle
5. Learn to document architectures

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
<u>1</u>	<u>Software architecture and its importance</u>	1	3
<u>2</u>	<u>Contexts of software architectures</u>	1	3
<u>3</u>	<u>Understanding Quality attributes of software architecture</u>	2	6
<u>4</u>	<u>Architectural tactics and patterns</u>	2	6
<u>5</u>	<u>Architecture in the life cycle</u>	2	6
<u>6</u>	<u>Architecture and requirements-Designing an architecture</u>	2	6
<u>7</u>	<u>Documenting software architectures-Architecture implementation and testing</u>	2	6
<u>8</u>	<u>Architecture evaluation-Architecture in the cloud</u>	2	6
Total		14	42

Textbook:

- Software Architecture in Practice, by Len Bass, Paul Clements, and Rick Kazman, 3rded, Addison-Wesley, 2012

Essential references:

- Software Design: From Programming to Architecture, by Eric Braude, Wiley, 2004.