

SWE-401: Software Design and Architecture Syllabus

General Information

Course Number	SWE-401
Credit Hours	3+0 (Theory Credit Hour = 3, Lab Credit Hours = 0)
Prerequisite	SWE-250: Software Engineering
Course Coordinator	Not specified

Course Objectives

This course is designed to give a breadth of software design and software architecture to help students understand the importance of architecture in developing software.

Students will be given a foundation for understanding existing software architectures and for designing and modeling architectures for real time systems with special focus on architecture's implementation, deployment, and long-term adaptation as without these activities a software misses its potential.

Students will also be given understanding and implementation knowledge of existing design patterns used in architectures that play major role for providing solutions to recurring problems.

Catalog Description

SWE-401

No.	Course Learning Outcome	Domain	Level	Assessment Tool
C1	To understand the role of software architecture, architectural styles & software design principles.	C	2	Assignments, Exams
C2	To understand design patterns for developing system software architecture.	C	2	Assignments, Exams
C3	To apply the use of tools for software architecture and design	C	3	Assignments, Exams

Domains: C=Cognitive, A=Affective, P=Psychomotor

Levels:

Cognitive = {1: Remembering, 2: Understanding, 3: Applying, 4: Analyzing, 5: Evaluating, 5: Creating }

Affective = {1: Receiving, 2: Responding, 3: Valuing, 4: Organizing, 5: Characterizing }

Psychomotor= {1: Imitation, 2: Manipulation, 3: Precision, 4: Articulation, 5: Naturalization }

Assessment criteria

S.No.	Assessment Task (examination, quizzes, assignments, presentations, etc.)	Proportion of Final Assessment/Marks
1.	Final exam	50
2.	Mid-Term	30
3.	Quiz	5
4.	Assignment/presentation	10
5.	Class participation/activities	5

Course Content

Week No.	Topics	Suggested Readings (Books)	CLO
1	1. Introduction 1.1 Introduction to Software Design & Software Architecture 1.2 Bridging Requirements & Implementation 1.3 Key software design principles 1.4 Architectural Styles 1.5 Quality Attributes 1.6 Software design guidelines	Text Book 1 Chapter 1	C1
2	2. Software Architecture Design Space 2.1. Types of Software Structures 2.1.1. Static Structure 2.1.2. Runtime(Dynamic) Structure 2.1.3. Management Structure 2.2. Building blocks of software architecture 2.2.1. Components, connectors and constraints 2.3. Software Connectors 2.3.1. Roles of the Connectors 2.3.2. Type of Connectors 2.3.2.1. Blocking & Non-blocking Connector 2.3.2.2. Initiator Connector 2.3.2.3. Information Carrier Connector 2.3.2.4. Signature based & Protocol based connector	Text Book 1 Chapter 2	C1
3	3. Models for Software Architecture 3.1. UML for Software Architecture 3.1.1. Structural Diagrams 3.1.2. Behavioral Diagrams 3.2. Architecture View Models 3.2.1. The Scenario View 3.2.2. The Logical or Conceptual View 3.2.3. The Development or Module View 3.2.4. The Process View 3.2.5. The Physical View 3.2.6. The User Interface View 3.2.7. Architecture Description Languages (ADL)	Text Book 1 Chapter 3	C3
4	5. Data Flow Architecture (DFA) 5.1. Application domain of DFA 5.2. Benefits and limitations of DFA 5.3. Batch sequential (Connectors, components, constraints and examples) 5.3.1. Benefits and limitations 5.4. Pipe and filter (Connectors, components, constraints and examples) 5.4.1. Active pipes and passive filters example 5.4.2. Active filters and passive pipes example 5.4.3. Benefits & limitations 5.4.4. Applicable domains	Text Book 1 Chapter 5	C1&C3
5	6. Introduction to Data centered architecture 6.1. Building blocks of data centered architecture 6.2. Repository architectural style 6.2.1. Overview 6.2.2. Benefits & limitations	Text Book 1 Chapter 6	C1&C3

	6.2.3.Applicable domains 6.3. Black board architectural style 6.3.1.Overview 6.3.2.Benefits & limitations 6.3.3.Applicable domains		
6-7	7. Hierarchy architecture 7.1. Introduction of hierarchical style 7.2. Main and sub routine architectural style 7.2.1.Benefits & limitations 7.3. Master-Slave architectural style 7.3.1.Applicable domains 7.4. Layered architectural style 7.4.1.SOAP example 7.4.2. OSI layer 7.5. Virtual machine architectural style 7.5.1.JVM architecture 7.5.2.Benefits & limitations 7.5.3.Applicable domains	Text Book Chapter 7	C1&C3
8	9. Interaction-oriented architecture 9.1. MVC architectural style 9.1.1. Java MVC 1 9.1.2. Java MVC 2 9.2. Framework based MVC architectural styles 9.2.1.Benefits & limitations of MVC Architecture 9.2.2.Applicable domains of MVC Architecture	Text Book 1 Chapter 9	C1&C3
9	10. Distributed architecture 10.1. Client server architecture as distributed architecture 10.1.1. Advantage & disadvantages 10.2. Broker architecture 10.2.1. Examples include RMI & CORBA 10.2.2. RMI: with example 10.2.3. CORBA: with example 10.2.4. Advantages & disadvantages of Broker Architecture	Text Book 1 Chapter 10	C1&C3
Mid-term Examination			
10-11	<ul style="list-style-type: none"> ▪ Introduction to design patterns <ul style="list-style-type: none"> ○ Design patterns and Frameworks ▪ Creational Design patterns <ul style="list-style-type: none"> ○ Factory Method design pattern with example <ul style="list-style-type: none"> ▪ Applicability ▪ Pros & Cons ○ Abstract Factory design pattern with example <ul style="list-style-type: none"> ▪ Applicability ▪ Pros & Cons ○ Builder design pattern with example <ul style="list-style-type: none"> ▪ Applicability ▪ Pros & Cons 	Text Book 2 Page 69-103	C2
12	<ul style="list-style-type: none"> ○ Prototype design pattern with example <ul style="list-style-type: none"> ▪ Applicability ▪ Pros & Cons ○ Singleton design pattern with example <ul style="list-style-type: none"> ▪ Applicability ▪ Pros & Cons 	Text Book Page 122-136	C2

13-14	Structural design patterns <ul style="list-style-type: none"> ○ Adapter pattern with example <ul style="list-style-type: none"> ▪ Applicability ▪ Pros & Cons ○ Bridge pattern with example <ul style="list-style-type: none"> ▪ Applicability ▪ Pros & Cons ○ Decorator pattern with example <ul style="list-style-type: none"> ▪ Applicability ▪ Pros & Cons ○ Composite pattern with example <ul style="list-style-type: none"> ▪ Applicability ▪ Pros & Cons 	Text Book 2 Page 146-233	C2
15-16	Behavioral design patterns <ul style="list-style-type: none"> ○ Command pattern with example <ul style="list-style-type: none"> ▪ Applicability ▪ Pros & Cons ○ Iterator pattern with example <ul style="list-style-type: none"> ▪ Applicability ▪ Pros & Cons ○ Observer pattern with example <ul style="list-style-type: none"> ▪ Applicability ▪ Pros & Cons ▪ 	Text Book 2 Page 268-336	C2
Final-term Examination			

CLO-PLO Map

CLO ID	Graduate Attribute (PLOs)											
	GA1	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9	GA10	GA11	GA12
CLO 1	1	0	0	0	0	0	0	0	0	0	0	0
CLO 2	1	0	0	0	0	0	0	0	0	0	0	0
CLO 3	0	0	0	0	1	0	0	0	0	0	0	0

Text Book

<ol style="list-style-type: none"> 1. Software Architecture and Design Illuminated By: k. Qian, j. & B. Publishers 2. Dive Into Design Patterns By: Alexander Shvets (2019)

Reference Material

<ol style="list-style-type: none"> 1. Software Architecture in Practice By: Len Bass, Paul Clements, Rick Kazman, & Addison Wesley, Second Edition

Approvals

Prepared By	Sana Fatima
Approved By	Not Specified
Last Update	8/11/2023

Computer Science/Software Engineering Program Learning Outcomes

GA: Graduate Attributes

GA1 Computing Knowledge: An ability to apply knowledge of mathematics, science, computing fundamentals and computing specialization to the solution of complex computing problems.

GA2 Problem Analysis: An ability to identify, formulate, research literature, and analyze complex computing problems reaching substantiated conclusions using first principles of mathematics, natural sciences and computing sciences.

GA3 Design/Development of Solutions: An ability to design solutions for complex computing problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.

GA4 Investigation: An ability to investigate complex computing problems in a methodical way including literature survey, design and conduct of experiments, analysis and interpretation of experimental data, and synthesis of information to derive valid conclusions.

GA5 Modern Tool Usage: An ability to create, select and apply appropriate techniques, resources, and modern IT tools, including prediction and modeling, to complex computing activities, with an understanding of the limitations.

GA6 The Computer Scientist and Society: An ability to apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional computing practice and solution to complex computing problems.

GA7 Environment and Sustainability: An ability to understand the impact of professional computing solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.

GA8 Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of computing practice.

GA9 Individual and Team Work: An ability to work effectively, as an individual or in a team, on multifaceted and /or multidisciplinary settings.

GA10 Communication: An ability to communicate effectively, orally as well as in writing, on complex computing activities with the computing community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

GA11 Project Management: An ability to demonstrate management skills and apply computing principles to one's own work, as a member and/or leader in a team, to manage projects in a multidisciplinary environment.

GA12 Lifelong Learning: An ability to recognize importance of, and pursue lifelong learning in the broader context of innovation and technological developments