Adama Science and Technology University

1	College: **Electrical Engineering and Computing**	Department: **Computer Science and Engineering**				
2	Course Category	***Core Module***				
	Course Name	***Software Architecture and Design***				
	Course Code:	***SEng3204***				
3	Synopsis:	In this course, An in-depth look at the software design. Continuation of the study of design patterns, frameworks, and architectures. Survey of current middleware architectures and technologies. Design of a new systems using middleware. Component based design. Measurement theory and appropriate use of metrics in design. Designing for qualities such as performance, safety, security, reusability, reliability, etc. Measuring internal qualities and complexity of software.				
4	Name(s) of Academic Staff:	Tefera Y.				
5	Semester and Year offered:	Semester:	II	Year:	3	
6	Credit Hour:	3				
7	Prerequisite/ Co-requisite: (if any)	Software Requirements engineering (SEng3201)				
8	Course Learning Outcome (CLO): At the end of the course the student will be able to do:					
	CLO1	Discuss basic design concepts and principles in software development				
	CLO2	Applying a variety of design concepts and architectural styles in designing a wide variety of software solutions.				
	CLO3	Using the sound quality metrics as objectives for designs, and then measuring and assessing designs to ensure the objectives have been meet.				
	CLO4	Differentiate design quality attributes during software development				
	CLO5	Apply system decomposition and sub system integration techniques for large scale software system				
9	Mapping of the course Learning Outcomes to the program Learning Outcomes, Teaching Methods and Assessment:					

Course Learning Outcomes (CLO)	Program Learning Outcomes (PO)														
										Assessment					
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	Teaching Methods	Test	Quiz	Mid	Assi	Proj ect	Final

										L	T	P	O						
	CLO1									√			√			√	√		√
	CLO2									√					√	√			√
	CLO3						√		√	√						√	√		√
	CLO4								√	√				√			√		√
	CLO5			√						√								√	√
	Indicate the relevance between the CLO and PO by ticking "√"on the appropriate relevant box																		

10	Transferable Skills (if applicable) (Skills learned in the course of study which can be useful and utilized in other settings)	
	1	
	2	
	3…etc.	

11	Distribution of Student Learning Time (SLT)								
	Course Content Outline	CLO	Teaching and Learning Activities						Total (SLT)
			Guided learning (F2F)				Guided Learning (NF2F)	Independent Learning (NF2F)	
			L	T	P	O			
	Chapter 1: Introduction 1.1 Architectural Thinking 1.2 Modularity 1.3 Architecture Characteristics Defined 1.4 Identifying Architectural Characteristics 1.5 Measuring and Governing Architecture Characteristics 1.6 Scope of Architecture Characteristics 1.7 key issue in Software Design 1.7.1 Concurrency 1.7.2 Control and Handling of Events 1.7. 3. Data Persistence 1.7.4 Distribution of Components 1.7.5 Error and Exception Handling and Fault Tolerance 1.7.6 Interaction and Presentation 1.7.7 Security	CLO1	6					7	10hr
	Chapter 2: Architecture Styles	CLO2	4					5	7hr

	2.1 Layered Architecture Style 2.2 Pipeline Architecture Style 2.3 Microkernel Architecture Style 2.4 Client/Server Architecture Style 2.5 Event-Driven Architecture Style 2.6 Space-Based Architecture Style 2.7 Orchestration-Driven Service-Oriented Architecture 2.8 Micro services Architecture 2.9 architecture in the Edge/ cloud 2.10 Choosing the Appropriate Architecture Style								
	Chapter 3: Subsystem Design	CLO5	8					8	10hr
	3.1 Definition 3.2 Modeling Subsystems, Interfaces, and Layers 3.2.1 Subsystem Interface Dependency Viewpoint 3.2.1.1 External Interface Design 3.2.1.2 Internal Interface Design 3.2.1.3 Graphical User Interface Design 3.2.2 Enhancing the Subsystem Dependency Views with layers 3.2.3 Top-level Dependencies 3.2.4 The Layered Subsystem Viewpoint 3.3 Mapping Subsystems and Layers to Implementation 3.3.1 Subsystems, layers, and build trees 3.3.2 Subsystems and components								
	Chapter 4: Transaction and Data Design	CLO3	5					8	8hr
	4.1 Logical Data Architecture 4.1.1 Logical data model stability 4.1.2 Effects of the stable logical data model 4.2 Logical Data Viewpoint 4.2.1 Logical Data View example 4.2.2 Logical Data View for messaging 4.3 Data Model Design – Other Considerations 4.3.1 Data models and layers 4.3.2 Data models and reflection 4.3.3 Mapping objects to relational database 4.4 Transaction Design 4.4.1 Transaction concepts 4.4.2 Modeling transaction dynamics 4.4.3 Transactions and interface design								

	Chapter 5: Software Architecture Process	CLO2	2						6	
	5.1 Process Outline 5.1.1 Determine Architectural Requirements 5.1.2 Identifying Architecture Requirements 5.1.3 Prioritizing Architecture Requirements 5.2 Architecture Design 5.2.1 Choosing the Architecture Framework 5.2.2 Allocate Components 5.3 Validation 5.3.1 Using Scenarios 5.3.2 Prototyping 5.4 Design Quality Attribute 5.4.1 availability 5.4.2 Interoperability 5..4.3 Performance 5.4.4 Security 5.4.5 Testability									
	Chapter 6: Introduction to Middleware Architectures and Technologies	CLO5	5						6	
	6.1 Middleware Technology Classification 6.2 Distributed Objects 6.3 Message-Oriented Middleware									
	Total		42						38	79hrs

Assessment					
Continuous Assessment		% Total-60(%)	F2F	NF2F	SLT
1	**Assignment I**	10%	1	6	5
2	**Test/quiz**	10%	2		1
5	**Project**	15%	2	18	20
6	**Mid Exam**	25%	1	8	10
				Total	**36hr**
Final Exam		Percentage 40 (%)	F2F	NF2F	SLT
Final Exam		40%	3	12	15
				Grand Total SLT	**120hr**

	L = Lecture, T = Tutorial, P = Practical, O = Others, F2F = Face to Face, NF2F = Non Face to Face Note: indicates the CLO based on the CLO's numbering in item 9.		
12	Special requirements and resources to deliver the course (e.g. software, computer lab, simulation room …etc.)	1	Choose an item.
		2	Choose an item.
		3	Choose an item.
		4	Choose an item.
		5	Choose an item.
13	Text book and reference: (note: ensure the latest edition /publication)	1	● Jeff Garland , Richard Anthony ,Large-Scale Software Architecturer,2003
		2	● Len Bass Paul Clements Rick Kazman,Software Architecture in Practice,2013
		3	● Ian Gorton,Essential Software Architecture,2011