



3 CHrs. – Three-Credit Hours

# Course Specification

— (Bachelor)

**Course Title:** System Integration and Architecture

**Course Code:** CIT 1401

**Program:** Bachelor in Information Technology

**Department:** Information Technology

**College:** Faculty of Computers and Information Technology

**Institution:** University of Tabuk

**Version:** 1.0

**Last Revision Date:** 27 July 2022



## Table of Contents

<b>A. General information about the course:</b> .....	3
<b>B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods</b> .....	4
<b>C. Course Content</b> .....	4
<b>D. Students Assessment Activities</b> .....	5
<b>E. Learning Resources and Facilities</b> .....	5
<b>F. Assessment of Course Quality</b> .....	5
<b>G. Specification Approval</b> .....	6



## A. General information about the course:

### 1. Course Identification

<b>1. Credit hours:</b>					
3 CHrs. (Three-Credit Hours)					
<b>2. Course type</b>					
A.	<input type="checkbox"/> University	<input type="checkbox"/> College	<input checked="" type="checkbox"/> Department	<input type="checkbox"/> Track	<input type="checkbox"/> Others
B.	<input checked="" type="checkbox"/> Required		<input type="checkbox"/> Elective		
<b>3. Level/year at which this course is offered: (7/4)</b>					
<b>4. Course general Description:</b>					
This course studies the process of integrating different systems and software applications by examining current and emerging trends, strategies, and techniques for developing systems integration solutions effectively.					
<b>5. Pre-requirements for this course (if any):</b>					
Software Engineering (CSC 1301)					
<b>6. Co-requisites for this course (if any):</b>					
<b>7. Course Main Objective(s):</b>					
<ul style="list-style-type: none"> <li>• Understand the up-front need to factor in the integration, test, verification and validation of a system in determining the system requirements and architecture.</li> <li>• Review and evaluate the various system integration, test, verification, and validation models in use today.</li> <li>• Plan the integration, verification and validation of the system of interest being developed.</li> <li>• Understand the criticality of defining, documenting and managing interfaces during system development.</li> <li>• System engineer and architect the integration and test environment.</li> <li>• Select suitable test methods, techniques, and metrics.</li> <li>• Apply fault diagnosis techniques.</li> <li>• Adopt a systems perspective when making integration and test decisions that affect the determination of performance, development time, or total ownership cost of the system.</li> </ul>					

### 2. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	45	100



No	Mode of Instruction	Contact Hours	Percentage
2	E-learning		
3	Hybrid <ul style="list-style-type: none"> <li>Traditional classroom</li> <li>E-learning</li> </ul>		
4	Distance learning		

### 3. Contact Hours (based on the academic semester)

No	Activity	Contact Hours
1.	Lectures	45
2.	Laboratory/Studio	
3.	Field	
4.	Tutorial	
5.	Others (specify)	
<b>Total</b>		<b>45</b>

## B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Code of PLOs aligned with program	Teaching Strategies	Assessment Methods
<b>1.0</b>	<b>Knowledge and understanding</b>			
1.1	Recognize the requirement engineering and its challenges.	K1,K2	Presentation Using case study to extract requirement for a system	<ul style="list-style-type: none"> <li>Home works</li> <li>Project</li> <li>Exam</li> </ul>
1.2	Describe and understand the importance of software documentation.	K3	Presentation Using case study to extract requirement for a system	<ul style="list-style-type: none"> <li>Home works</li> <li>Project</li> <li>Exam</li> </ul>



Code	Course Learning Outcomes	Code of PLOs aligned with program	Teaching Strategies	Assessment Methods
<b>2.0 Skills</b>				
2.1	Write software document.	S2	<p>Presentation</p> <p>Assign team project for the team under instructor supervision</p> <p>Class discussion</p> <p>Tutorial on the tool</p>	<ul style="list-style-type: none"> <li>• Home works</li> <li>• Project</li> <li>• Exam</li> </ul>
2.2	Plan to achieve software with error-free code.	S2,S3	<p>Presentation</p> <p>Assign team project for the team under instructor supervision</p> <p>Class discussion</p> <p>Tutorial on the tool</p>	<ul style="list-style-type: none"> <li>• Home works</li> <li>• Project</li> <li>• Exam</li> </ul>
2.3	Demonstrate the requirement engineering.	S4	<p>Presentation</p> <p>Assign team project for the team under instructor supervision</p> <p>Class discussion</p> <p>Tutorial on the tool</p>	<ul style="list-style-type: none"> <li>• Home works</li> <li>• project</li> </ul>
2.4	Illustrate and understand the problems and challenges of acquiring, integrate and implementing a software package.	S2	<p>Presentation</p> <p>Assign team project for the team under instructor supervision</p> <p>Class discussion</p> <p>Tutorial on the tool</p>	<ul style="list-style-type: none"> <li>• Home works</li> <li>• Project</li> <li>• Exam</li> </ul>



Code	Course Learning Outcomes	Code of PLOs aligned with program	Teaching Strategies	Assessment Methods
<b>3.0</b>	<b>Values, autonomy, and responsibility</b>			
3.1	Demonstrate and present project to various audiences.	V2	Class discussion	Project
3.2	Discuss requirement engineering with stakeholders.	V2	Class discussion	Project

### C. Course Content

No	List of Topics	Contact Hours
1.	Systems Engineering Overview(Part 1): System procurement	3
2.	Systems Engineering Overview (Part 2): Systems development	3
3.	Requirements Validation (Part 1): Requirements engineering	3
4.	Requirements Validation (Part 2): Verification and Validation	3
5.	System Interfaces (Part 1): API and RESTful services	3
6.	System Interfaces (Part 2): Service engineering and Service composition	3
7.	System Architecture Validation & patterns design (Part 1): Architectural design and Methods to validate a software architecture	
8.	System integration (Part 1): Systems of systems, System complexity and Systems of systems classification	3
9.	System integration (Part 2): Reductionism and complex systems, Systems of systems engineering	3
10.	System integration (Part 3): Systems of systems architecture	3
11.	White Box Test Cases (Part 1): What is testing? and Test Techniques	3
12.	White Box Test Cases (Part 2): Flow Graph and loop testing	3
13.	Black Box Test Cases (Part 1): Black-box Testing Categories, Questions answered by Black-box Testing and Boundary Value Analysis.	3
14.	Fault Diagnosis: Fault-tolerance concepts, Software safety, Fault detection, Fault avoidance and Fault tolerance	3
15.	System transition: The Rational Unified Process and System Transition Phase	3
<b>Total</b>		<b>45</b>



## D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Participation	Throughout semester	10%
2.	Homework	4,6	10%
3.	Project	13	20%
	Midterm exam 1	6 or 7	10%
	Midterm exam 2	11 or 12	10%
	Final Exam	17-19	40%

\*Assessment Activities (i.e., Written test, oral test, oral presentation, group project, essay, etc.).

## E. Learning Resources and Facilities

### 1. References and Learning Resources

<b>Essential References</b>	<ul style="list-style-type: none"> <li>Stephen R. Schach, Object-Oriented Software Engineering, McGraw-Hill, 2007, ISBN-13: 9780073523330, ISBN: 007352333X.</li> <li>Ian Sommerville, Software Engineering, Boston: Pearson, 9 Edition, ISBN 0-13-705346-0, 2011.</li> <li>Andrew P. Sage, William B. Rouse, Handbook of Systems Engineering and Management, Wiley-Interscience, 2 Edition, 2009.</li> </ul>
<b>Supportive References</b>	<ul style="list-style-type: none"> <li>Nick Rozanski and Eóin Woods, Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives, Addison-Wesley Professional, 2 Edition, 2011.</li> <li>Roger S. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill Science/Engineering/Math, 7 Edition, 2009.</li> <li>Cem Kaner, et al., Testing Computer Software, Wiley; 2 Edition, 1999.</li> </ul>
<b>Electronic Materials</b>	
<b>Other Learning Materials</b>	

### 2. Required Facilities and equipment

Items	Resources
<b>facilities</b> (Classrooms, laboratories, exhibition rooms, simulation rooms, etc.)	<b>Lecture room of each section to accommodate 25 students.</b>
<b>Technology equipment</b> (projector, smart board, software)	<b>Projectors, data show, smart board, etc.</b>
<b>Other equipment</b> (depending on the nature of the specialty)	



## F. Assessment of Course Quality

Assessment Areas/Issues	Assessor	Assessment Methods
Effectiveness of Teaching	Faculty, Program Leaders, and Advisory Board	Both Direct and Indirect
	Students	Indirect
Effectiveness of Students Assessment	Faculty, Program Leaders, Advisory Board, and Independent Opinion	Both Direct and Indirect
Quality of Learning Resources	Faculty, Students, and Advisory Board	Indirect
The Extent to which CLOs have been Achieved	Faculty, Program Leaders, Advisory Board, and Independent Opinion	Direct (as in section B) and Indirect/Surveys
	Students	Indirect
Other	-	-

**Assessors** (Students, Faculty, Program Leaders, Peer Reviewer, Others (specify))

**Assessment Methods** (Direct, Indirect)

## G. Specification Approval

<b>COUNCIL /COMMITTEE</b>	
<b>REFERENCE NO.</b>	
<b>DATE</b>	

