National University of Sciences & Technology (NUST)
School of Electrical Engineering and Computer Science (SEECS)
Department of Software Engineering

Software Engineering

Course Code:	SE-200	**Semester:**	3rd
Credit Hours:	3+0	**Prerequisite Codes:**	CS-100: Fundamentals of ICT
Instructor:	**Zain ul Hassan Khan**	**Class:**	BESE-5B
Office:	TBA	**Telephone:**	03345383987
Lecture Days:	Monday and Thursday	**E-mail:**	**z.hassan@seecs.edu.pk,zainulhassan@gmail.com**
Class Room:	SEECS CR-08	**Consulting Hours:**	TBA
Knowledge Group:	Software Engineering	**Updates on LMS:**	Before every lecture

Course Description:

Software Engineering the application of a systematic, disciplined, quantifiable approach to the design, development, operation, and maintenance of software, and the study of these approaches. This course covers introductory topics in Software Engineering. Students are expected to have sound programming background before taking this course. It is designed to familiarize students with the fundamental concepts of software engineering, including: *Software Process Models, Agile Software Development, Requirements Engineering, System Modeling, Architectural Design, Design patterns, Software Testing, Software Evolution, software reuse and software project Management*.

This course will combine theoretical underpinnings of the software engineering fundamentals with the practical aspects of applied software engineering focusing on the main phases of software development life cycle. Students will be able to understand the role and scope of requirements engineering and know how to apply appropriate methods, techniques and tools to elicit, document and manage requirements. They will be able to define a system that satisfies the requirements. It will also educate the students with essential concepts of ethical software engineering and best practices. This course further aims to equip students with the necessary skills of system modeling using UML modeling language and CASE tools; capture requirements of a real system; build prototypes to specify, design and test system and gain practical experience through a semester long project activity.

Course Objectives:

This course introduces fundamentals of software engineering. The aim of the course is to study various software development models and phases of software development life cycle. The concepts of project management, change control, process management, software development and testing are introduced through hands-on Team Projects. To understand the role and scope of requirements engineering and know how to apply appropriate methods, techniques and tools to elicit, document and manage requirements. To be able to define a system that satisfies the requirements.

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**PLO**	**BT Level***
1. Understand the software process models and able to select the most appropriate one according to the situation.	1, 2	C-3, A-2
2. Construct a number of different UML analysis & design models and Implement the design using most appropriate architectural pattern	2, 3, 4, 5	C-4
3. Develop a software testing, evolution and maintenance strategy following a structured and organized process.	3, 5	C-3, 4
4. Learn to work in a team with understanding of ethical and professional issues.	8, 9, 10	C-6, A-5
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Mapping of CLOs to Program Learning Outcomes

PLOs/CLOs	CLO1	CLO2	CLO3	CLO4
PLO 1 (Engineering Knowledge)	√			
PLO 2 (Problem Analysis)	√	√		
PLO 3 (Design/Development of Solutions)		√		
PLO 4 (Investigation)			√	
PLO 5 (Modern tool usage)		√	√	
PLO 6 (The Engineer and Society)				
PLO 7 (Environment and Sustainability)				
PLO 8 (Ethics)				√
PLO 9 (Individual and Team Work)				√
PLO 10 (Communication)				√
PLO 11 (Project Management)				√
PLO 12 (Lifelong Learning)				

Mapping of CLOs to Assessment Modules and Weightages (In accordance with NUST statutes)

To be filled in at the end of the course.

Assessments/CLOs	CLO1	CLO2	CLO3	CLO4
Quizzes: 10%				
Assignments: 10%				
OHT-1: 15%				
OHT-2: 15%				
Project:10%				
End Semester Exam:40%				
Total : 100 %				

Books:	
Text Book:	1. Ian Sommerville Software Engineering, 9th ed. Addison-Wesley, 2010
Reference Books:	2. R.S. Pressman, Software Engineering: A Practitioner's Approach, 7th ed., McGraw-Hill Book Co., NY, 2010, ISBN 978-00-7337597-7

Topics to be Covered:

1. Introduction to Software Engineering, Evolving Role of Software, Problems in Software Industry
2. Software Process Models, Process Activities**,** Rational Unified Process.
3. Agile Software Development, Agile Methods, Extreme Programming, Agile Project Management,
4. Requirement Engineering, Requirements Specification, Elicitation, Analysis, Validation and Management.
5. Analysis modeling which includes Context Model, Structural Model, Interaction Model and Behavioral Model.
6. Software Architecture, Architectural Trade-off Analysis, Mapping Requirements into Architecture, Architectural Patterns, Software Design Principals.
7. Software Testing, Development Testing, Release testing, User Testing, Software Evolution, Software maintenance, Software Reuse

Week No.	Topics
1	**Introduction to Software Engineering** ✧ Course orientation ✧ Introduction ✧ Professional software development ✧ Software engineering ethics
2	**Software Processes** ✧ Software process models ✧ Process activities ✧ Coping with change ✧ The rational unified process
3	**Agile Software Development** ✧ Agile methods ✧ Plan-driven and agile development ✧ Extreme programming ✧ Agile project management
4	**Requirements Engineering** ✧ Functional and non-functional requirements ✧ The software requirements document ✧ Requirements specification
5	**Requirements Engineering (continued)** ✧ Requirement engineering processes ✧ Requirements elicitation and analysis ✧ Requirements validation ✧ Requirements management
6	**OHT-1**
7	**System Modeling** ➢ Tutorial: Modeling with UML ➢ Context models ➢ Interaction models
8	**System Modeling (continued)** ✧ Structural models ✧ Behavioral models ✧ Model-driven engineering

9	**Architectural Design** ✧ Architectural design decisions ✧ Architectural Views
10	**Architectural Design (continued)** ✧ Architectural Patterns ✧ Application Architectures
11	**System Design and Implementation** ✧ Object-oriented design using the UML ✧ Design Patterns
12	**OHT-2**
13	**Software Testing** ✧ Development testing ✧ Test-driven development
14	**Software Testing** ✧ Release testing ✧ User testing
15	**Component based Software Engineering** ✧ Components and component models ✧ CBSE processes ✧ Component Composition ✧ **Software Reuse**
16	**Software Evolution** ✧ Evolution processes ✧ Software maintenance **Project Management** ✧ Risk Management ✧ Managing People ✧ Teamwork
17	**Project Presentations**
18	Week 18: ESE

Tools / Software Requirement:
Start UML, Sparx Enterprise

Grading Policy:	
Quiz Policy:	The quizzes will be unannounced and normally last for ten minutes. The question framed is to test the concepts involved in last few lectures. Number of quizzes that will be used for evaluation is at the instructor's discretion.
Project Policy:	Students will be required to develop a project during the course which should be completed at the end of the semester. They will be graded based on project deliverables and presentation at the end. Students will work in a group/team for projects. A group of 3 students is recommended. At most 4 students are allowed. A team cannot be broken down during the semester.
Assignment Policy:	In order to develop comprehensive understanding of the subject, assignments will be given. Late assignments will not be accepted / graded. All assignments will count towards the total (No 'best-of' policy). The students are advised to do the assignment themselves. Copying of assignments is highly discouraged and violations will be dealt with severely by referring any occurrences to the disciplinary committee. The questions in the assignment are meant to be challenging to give students confidence and extensive knowledge about the subject matter and enable them to prepare for the exams.
Plagiarism:	SEECS maintains a zero tolerance policy towards plagiarism. While collaboration in this course is highly encouraged, you must ensure that you do not claim other people's work/ ideas as your own. Plagiarism occurs when the words, ideas, assertions, theories, figures, images, programming codes of others are presented as your own work. You must cite and acknowledge all sources of information in your assignments. Failing to comply with the SEECS plagiarism policy will lead to strict penalties including zero marks in assignments and referral to the academic coordination office for disciplinary action.