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:**	5th Semester
Credit Hours:	3+0	**Prerequisite Codes:**	CS-100: Fundamentals of ICT
Instructor:	**Ms. Ayesha Kanwal**	**Class:**	BSCS-6AB
Office:	A-205	**Telephone:**	
Lecture Days:	Mon, Wed, Thurs	**E-mail:**	**Ayesha.kanwal@seecs.edu.pk**
Class Room:	RIMMS CR-21, CR-22	**Consulting Hours:**	Thursday 3pm – 4pm
Knowledge Group:	Software Engineering	**Updates on LMS:**	After 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):	**PLO**	**BT-level**
1. Understand key principles of and processes of software engineering	PLO1	C-2
2. Gather requirements for software systems	PLO2	C-4
3. Create software design models	PLO3	C-3, C-4
4. Evaluate system requirement and design using suitable approach	PLO4	C-5

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

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

Books:

Text Book:	1. Ian Sommerville Software Engineering, 10th ed. Addison-Wesley, 2013
Reference Books:	1. R.S. Pressman, Software Engineering: A Practitioner's Approach, 8th ed., McGraw-Hill Book Co., NY, 2010

Pre-Requisites:

Existing	CS-100: Fundamentals of ICT
Proposed	CS212 Object Oriented Programming

Lecture breakdown

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
3	**Agile Software Development** ✧ Agile methods

	✧ Plan-driven and agile development ✧ Extreme programming ✧ Agile project management
4	**Requirements Engineering I** ✧ Functional and non-functional requirements ✧ The software requirements document ✧ Requirements specification
5	**Requirements Engineering II** ✧ Requirement engineering processes ✧ Requirements elicitation and analysis ✧ Requirements validation ✧ Requirements management
6	**System Modeling I** ✧ Tutorial: Modeling with UML ✧ Context models
7	**OHT-1**
8	**System Modeling II** ✧ Interaction models ✧ Structural models ✧ Behavioral models ✧ Model-driven engineering
9	**System Modeling III** ✧ Behavioral models ✧ Model-driven engineering
10	**Architectural Design** ✧ Architectural design decisions ✧ Architectural Views ✧ Architectural Patterns
11	**Activity Week** ✧ **Case Study**: Requirement Engineering ✧ **Case Study**: System Modeling
12	**System Design and Implementation** ✧ Object Oriented Design using the UML ✧ Design Patterns
13	**OHT-2**
14	**Software Testing I** ✧ Development Testing ✧ Test-driven Development
15	**Software Testing II** ✧ Release testing ✧ User testing
16	**Software Evolution and Advanced Topics** ✧ Evolution processes ✧ Software change and maintenance **Advanced Topics (**Requirement and Modeling of distributed Systems, Software engineering for adaptive systems, mobile computing**)**
17	**Advanced Topics (**Requirement and Modeling of distributed Systems, Software engineering for adaptive systems, mobile computing**)**
18	Week 18: ESE

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.

Program Learning Outcomes (PLOs)

Program outcomes are the narrower statements that describe what students are expected to know and be able to do by the time of graduation. These relate to the knowledge, skills and attitude that the students acquire while progressing through the program. The program must demonstrate that by the time of graduation the students have attained a certain set of knowledge, skills and behavioral traits, at least to some acceptable minimum level. Specifically, it is to be demonstrated that the students have acquired the following graduate attributes:

(i) **Engineering Knowledge:** An ability to apply knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

(ii) **Problem Analysis:** An ability to identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.

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

(iv) **Investigation:** An ability to investigate complex engineering 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.

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

(vi) **The Engineer 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 engineering practice and solution to complex engineering problems.

(vii) **Environment and Sustainability:** An ability to understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.

(viii) **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.

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

(x) **Communication:** An ability to communicate effectively, orally as well as in writing, on complex engineering activities with the engineering 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.

(xi) **Project Management:** An ability to demonstrate management skills and apply engineering principles to one’s own work, as a member and/or leader in a team, to manage projects in a multidisciplinary environment.

(xii) **Lifelong Learning:** An ability to recognize importance of, and pursue lifelong learning in the broader context of innovation and technological developments.