

TED UNIVERSITY, COURSE SYLLABUS

Faculty	Engineering	Department	Software Engineering
----------------	-------------	-------------------	----------------------

Course Code & Number	SENG 214/CMPE313	Course Title	C-Software Engineering
Type of Course	<input checked="" type="checkbox"/> Compulsory <input type="checkbox"/> Elective	Semester	<input type="checkbox"/> Fall <input checked="" type="checkbox"/> Spring <input type="checkbox"/> Summer
Course Credit Hours	(3+0+0) 3	Number of ECTS Credits	6
Pre-requisite	CMPE 211 or CMPE 114	Co-requisite	N/A
Mode of Delivery	<input checked="" type="checkbox"/> Face-to-face <input type="checkbox"/> Distance learning	Language of Instruction	<input checked="" type="checkbox"/> English <input type="checkbox"/> Turkish
Course Coordinator	Dr. Elif KURTARAN ÖZBUDAK	Course Lecturer(s)	Dr. Elif KURTARAN ÖZBUDAK
Required Reading	<p><i>Software Engineering</i> by Ian Sommerville 10th Edition, Pearson Education Limited 2016, ISBN: 978-0-13-394303-0</p> <p>Coursera Courses by The Hong Kong University of Science and Technology :</p> <p>Course 1 - Software Engineering: Modeling Software Systems using UML</p> <p>Course 2 - Software Engineering: Implementation and Testing</p> <p>Course 3 - Software Engineering: Software Design and Project Management</p>	Recommended Reading	<p><i>Software Engineering: A Practitioner's Approach</i> by Roger Pressman and Bruce Maxim, 9th Edition, McGraw-Hill Education, ISBN-13: 978-1-260-54800-6.</p>
Course Catalog Description	<p>Software Engineering: introduction, basic terminology, principles and ethics. Software Processes: process models, activities. Agile Development: agile methodology, scrum. Software Requirements: eliciting requirements, developing use cases, modeling with scenario-based methods, modeling with class-based methods, UML models and sequence diagrams. Design Concepts: patterns, software architecture, and object-oriented design. Architectural Design: software architectures and styles. Software Quality: concepts, quality assurance, achieving software quality. Software Testing: strategies, conventional, object oriented application testing. Software Maintenance.</p>		

Course Objectives	The objective of this course is to provide software engineering knowledge and skills to participate in a large scale software development environment and manage a software development process. The students will be able to put the theoretical knowledge of software requirements elicitation, developing use cases, modelling, designing and quality assessment into practical use.			
Course Learning Outcomes	Upon successful completion of this course, the students will be able to: 1. Discuss software processes and development lifecycle, 2. Prepare requirements for a software, 3. Construct a software design, 4. Describe the concepts of software quality, 5. Prepare software testing scenarios.			
Learning Activities & Teaching Methods¹	<input checked="" type="checkbox"/> Brainstorming <input checked="" type="checkbox"/> Case Study/Scenario Analysis <input type="checkbox"/> Collaborating <input type="checkbox"/> Concept Mapping <input checked="" type="checkbox"/> Demonstrating <input checked="" type="checkbox"/> Discussions / Debates <input type="checkbox"/> Drama / Role Playing <input type="checkbox"/> Experiments <input type="checkbox"/> Field Trips <input type="checkbox"/> Guest Speakers	<input checked="" type="checkbox"/> Hands-on Activities <input type="checkbox"/> Inquiry <input type="checkbox"/> Microteaching <input checked="" type="checkbox"/> Oral Presentations / Reports <input type="checkbox"/> Peer Teaching <input checked="" type="checkbox"/> Predict-Observe-Explain <input checked="" type="checkbox"/> Problem Solving <input checked="" type="checkbox"/> Questioning <input checked="" type="checkbox"/> Reading	<input type="checkbox"/> Scaffolding / Coaching <input type="checkbox"/> Seminars <input type="checkbox"/> Service Learning <input type="checkbox"/> Simulations & Games <input checked="" type="checkbox"/> Telling / Explaining <input type="checkbox"/> Think-Pair-Share <input checked="" type="checkbox"/> Video Presentations <input type="checkbox"/> Web Searching <input checked="" type="checkbox"/> Other(s):Coursera.....	
Assessment Methods & Criteria²	<input type="checkbox"/> Case Studies / Homework <input type="checkbox"/> Lab Assignment <input type="checkbox"/> Observation <input type="checkbox"/> Oral Questioning <input type="checkbox"/> Peer Evaluation <input type="checkbox"/> Performance Project (Written, Oral) <input type="checkbox"/> Portfolio%%%%%%	<input type="checkbox"/> Presentation (Oral, Poster) <input checked="" type="checkbox"/> Project <input checked="" type="checkbox"/> Quiz <input type="checkbox"/> Self-evaluation <input checked="" type="checkbox"/> Test/Exam <input checked="" type="checkbox"/> Other(s): In class activities	(0 %) (25 %) (10 %)% (60 %) (5%)
Student Workload³	<input checked="" type="checkbox"/> Case Study Analysis <input checked="" type="checkbox"/> Course Readings <input type="checkbox"/> Debate <input type="checkbox"/> Demonstration <input checked="" type="checkbox"/> Exams/Quizzes <input type="checkbox"/> Field Trips/Visits <input checked="" type="checkbox"/> Hands-on Work <input type="checkbox"/> Lab Applications <input checked="" type="checkbox"/> Lectures	(20 hrs) (20 hrs) (... hrs) (... hrs) (30 hrs) (... hrs) (10 hrs) (... hrs) (42 hrs)	<input type="checkbox"/> Online Discussion <input checked="" type="checkbox"/> Oral Presentation <input type="checkbox"/> Poster Presentation <input type="checkbox"/> Report on a Topic <input type="checkbox"/> Research Review <input type="checkbox"/> Resource Review <input checked="" type="checkbox"/> Team Meetings <input type="checkbox"/> Web Designs <input type="checkbox"/> Work Placement	(... hrs) (10 hrs) (... hrs) (... hrs) (... hrs) (... hrs) (20 hrs) (... hrs) (... hrs)

¹ Multiple options possible.

² Multiple options possible. A percentage must be stated for the selected assessment method & criteria.

³ Multiple options possible. The student workload is found by multiplying the number and duration (hour) of the activity involved.

	<input checked="" type="checkbox"/> Mock Designs	(20 hrs)	<input type="checkbox"/> Workshop	(... hrs)
	<input type="checkbox"/> Observation	(... hrs)	<input checked="" type="checkbox"/> Other(s): Coursera...activities.....	(.10.. hrs)
	Total Workload⁴			182

COURSE ASSIGNMENTS
A. Mid-terms [25%]
There will be one midterm exam.
B. Quizes in Coursera & Certification [10%]
Coursera Certification: %10 (complete successfully all the quizzes at the end of each lecture of Coursera courses (3 courses) and get your certificate. Certification of Course-1: %4 , Course 2 and 3 is worth 3%.
C. In Class Activities / Active Class Participation [5%]
It is encouraged that you participate in class activities and discussions. Class participation is awarded based on your attendance to written class activities and your contribution to class discussion.
C. Project (25%)
Each project will be carried out in a group. Suggested project topics will be announced later on. Software development processes will be conducted in the project. Project hands-on works will be done/start in the class. Each group will submit only one deliverable, the points of the students will differentiate according to their hands-on performance in the class. At the end of the semester, each group will submit a prototype/mock-up design of their project. The project will be presented in the class, and presentation will be evaluated as well.
D. Final [35%]
There will be one final exam including the whole topics of course.

COURSE POLICIES
Attendance
Attending is NOT mandatory , but strongly recommended. Some hands-on activities and discussions will be done in the lectures. If you would like to collect points for these activities (see active class participation clause), you need to attend the lectures.
Missed Deadline/Exam

⁴ Computing the ECTS credits of a course: Total workload / 25 or 30 hours = ECTS credit and 1 ECTS credit = 25-30 hours

There will be only one make-up exam for midterm and final exam, if the student can provide a legal document confirming a health issue at the time of the exam, or with the consensus of the CMPE faculty.

Late Assignment Submission Policy

Late submissions will be graded with penalty. The allowed number of days for late submissions is maximum three, there will be a 20% decrease per day.

Extra Credit

Extra credits will not be offered.

Project Assignment Rules

All project assignment works will be done as a group, unless told differently. Project hands-on works will be done/start in the class. A group will submit only one report, the points of the students will differentiate according to their hands-on performance in the class.

At the end of the semester, each group will submit a mock-up design of their project.

Plagiarism

All of the following are considered plagiarism:

- turning in someone else's work as your own
- copying words or ideas from someone else without giving credit
- failing to put a quotation in quotation marks
- giving incorrect information about the source of a quotation
- changing words but copying the sentence structure of a source without giving credit
- copying so many words or ideas from a source that it makes up the majority of your work, whether you give credit or not"

Plagiarism is a very serious offense and will be penalized accordingly by the university disciplinary committee. The best way to avoid accidentally plagiarizing is to work on your own before you ask for the help of other resources.

Cheating

Cheating has a very broad description which can be summarized as "acting dishonestly". Some of the things that can be considered as cheating are the following:

- Copying answers on examinations, homework and laboratory works,
- Using prohibited material on examinations,
- Lying to gain any type of advantage in class
- Providing false, modified or forged data in a report
- Plagiarizing.
- Modifying graded material to be regraded.
- Causing harm to colleagues by distributing false information about an examination, homework or laboratory

Cheating is a very serious offense and will be penalized accordingly by the university disciplinary committee.

Class Readings

Class readings are necessary but not mandatory. The material covered in class by your instructor will only provide a fundamental understanding of the general context. The reading materials will be provided by the instructor, at the relevant week. In this course, we will apply the flipped learning

model by using Coursera courses. For a detailed information about flipped learning see [flipped-learning.pdf \(tedu.edu.tr\)](https://tedu.edu.tr/~flipped-learning.pdf). The material will also cover the Coursera courses as defined in the syllabus. For each week, before coming to classes you have to watch the corresponding Coursera lecture specified by your instructor and be prepared for the flipped course.

TENTATIVE COURSE OUTLINE					
Week	Dates	Topics	Readings	Coursera Material	Assignments
1	19.02.-24.02	Introduction to Software Engineering	Chapter 1		
2	26.02-01.03	Software Processes + IEEE 12207	Chapter 2	Course 1 – Week 1 Introduction to Software Engineering	Form your teams (01.03.2024)
3	4.03-8.03	Agile Development	Chapter 3	Course 2 - Week 2 Software Development Processes	Projects topics release (08.03.2024)
4	11.03-15.03	Requirements Engineering	Chapter 4		Assignment 1: Project Proposals due (15.03.2024)
5	18.03-22.03	FLIPPED CLASSROOM: System Modeling (COURSERA) C-HOA – Class Diagram Exercises C-HOA-Domain Model Exercise		Course 1 - Week 2: Modeling Software Systems Using UML Week 3 : Types of Relationships in Class Diagram Week 4 :System Requirements Capture and Domain Modeling	
6	25.03-29.03	FLIPPED CLASSROOM: Requirements Modeling (COURSERA)		Course 1 – Week 5 : Use Case Modeling Week 6 –Use Case Specification	

		C-HOA-Lecture5 - Domain Model Exercise C-HOA-Lecture 7 - Use Case Modeling Exercise C-HOA- Lecture 7 - System Requirements Capture Exercise		Week 7 :Non-Functional Requirements	
7	01.04-05.04	FLIPPED CLASSROOM: Design Concepts (COURSERA) C-HOA-System Design and Analysis C-HOA-Design Pattern Exercise		Course 3 - Week 1 : System Analysis and Design Week 2 :Design Patterns (Lecture 3)	Assignment 2: SRS due (05.04.2024)
8	FEAST OF RAMADAN				
9	15.04-19.04	FLIPPED CLASSROOM: Managing Software Development		Course 3- Week 3 Lecture 6 Managing SW Development	GET YOUR COURSERA CERTIFICATE FOR COURSE -1 (19.04.24)
10	22.04-26.04 (April 23, No Lecture for Sect 01)	Software Quality Management	Chapter 24		Midterm (April 26, 18:00) (No lecture for Sect03)
11	(29.04-3.05 May 1, No Lecture for Sect02)	FLIPPED CLASSROOM: Software Quality Assurance Configuration Management		Course 3- Week 3 Lecture 5 SQA Course 2- Week 4 Debugging and Configuration Management	
12	6.05-10.05	Software Testing	Chapter 8		Assignment 3: Design Document due (10.05.2024)
13	13.05-17.05	Software Maintenance FLIPPED CLASSROOM: Software Testing HOA - Lecture 7 - Acceptance Testing Exercise	Chapter 9	Course 2 - Week 5: White Box Testing Week 6 :Black Box Testing	

				Course 2 Week 7: Acceptance Testing	
14	20.05-24.05	Project Presentations			GET YOUR COURSERA CERTIFICATE FOR COURSE -2 and 3 (24.05.2024)
15	27.05-31.05	Project Presentations			
FINAL EXAM					

Prepared By⁵	Dr. Elif KURTARAN ÖZBUDAK	Date	19/02/2024
Revised By⁶		Rev. Date	

STUDENT SERVICES INFO:

Student Development and Psychological Counseling Center:

The Center is a service mandated with providing crisis intervention and supportive listening services to the campus community. A major part of fulfilling that mandate is raising awareness of our service so students know they are never alone in dealing with problems. You may contact the SDPCC at: ogrencidanismamerkezi@tedu.edu.tr, 0312 585 0316, Office A122, Or visit their website at <http://csc.tedu.edu.tr/>

TEDU COPeS - Psycho-Social Support

TED University Psychosocial Support Team was initially established in order to facilitate coping with the psychological, social, familial, academic, and professional difficulties that may arise due to adverse conditions associated with COVID-19 pandemic for TEDU students and employees.

In time we have expanded our services to provide psychosocial support in diverse disasters. In this line, TEDU COPeS offers psychosocial support for TED University students and employees in the aftermath of Kahramanmaraş earthquakes.

For further information and/or questions, visit their website at <https://copes.tedu.edu.tr/>

Specialized Support and Students with Disabilities

⁵ It is the first person to prepare the course profile form and the first preparation date.

⁶ It is the person who revised the course profile form and the date of revision. It will be used if the course profile form is revised. In the new course proposal, this field will be left blank.

Students who may require specialized support due to a disability affecting mobility, vision, hearing, learning, mental or physical health should consult with Specialized Support and Disability Coordinator, Asst. Prof. Emrah Keser E-mail: emrah.keser@tedu.edu.tr, or visit the website at <https://www.tedu.edu.tr/tr/main/engelsiz-tedu>

APPENDIX 1- Coursera Course List

Course 1: Software Engineering: Modeling Software Systems using UML	https://www.coursera.org/learn/software-engineering-modeling-software-systems-using-uml?specialization=software-engineering
Course 2: Software Engineering: Implementation and Testing	https://www.coursera.org/learn/software-engineering-implementation-and-testing
Course 3: Software Engineering: Software Design and Project Management	https://www.coursera.org/learn/software-engineering-software-design-and-project-management?specialization=software-engineering

[Course 1 - Software Engineering: Modeling Software Systems using UML](#)

Week 1 : Introduction to Software Engineering

1. [Video: Lecture 1-1 - Software Development is Complicated](#)
2. **Video:** Lecture 1-2 Dealing with the Complexity
3. **Video:** Lecture 1-3 What is Software Engineering
4. **Reading:** Lecture 1 - Introduction to Software Engineering

Graded: Quiz 1 - Introduction

Week 2: Modeling Software Systems Using UML

1. [Video: Lecture 2-1 Modeling Introduction](#)
2. **Video:** Lecture 2-2 UML Class Diagram
3. **Video:** Lecture 2-3 Association and Aggregation Relationships
4. **Reading:** Lecture 2 - Modeling Software Systems using UML
5. **Reading:** Lecture 2 - Class Diagram Exercise

Graded: Quiz 2 - Modeling Software Systems Using UML

Week 3 : Types of Relationships in Class Diagram

1. [Video: Lecture 3-1 Association Class](#)
2. **Video:** Lecture 3-2 Generalization
3. **Video:** Lecture 3-3 Class Diagram Summary
4. **Reading:** Lecture 3 - Modeling Software Systems using UML
5. **Reading:** Lecture 3 - Class Diagram Exercise 1
6. **Reading:** Lecture 3 - Class Diagram Exercise 2

Graded: Quiz 3 - Modeling Software Systems Using UML

Week 4 :System Requirements Capture and Domain Modeling

1. [Video: Lecture 4-1 - Introduction to System Requirements Capture](#)
2. **Video:** Lecture 4-2 - Domain Modeling - Evaluating Classes
3. **Video:** Lecture 4-3 - Domain Modeling - Evaluating Associations and Attributes
4. **Reading:** Lecture 4 - System Requirements Capture
5. **Reading:** Lecture 4 - Domain Model Exercise

Graded: Quiz 4 - System Requirements Capture

Week 5 : Use Case Modeling

1. [Video: Lecture 5-1 - Use Case Modeling - Actors](#)
2. **Video:** Lecture 5-2 - Use Cases
3. **Video:** Lecture 5-3 - Use Case Modeling Example
4. **Reading:** Lecture 5 - System Requirements Capture
5. **Reading:** Lecture 5 - Domain Model Exercise

Graded: Quiz 5 - System Requirements Capture

Week 6- Use Case Specification

1. [Video: Lecture 6-1 - Use Case Specification](#)
2. **Video:** Lecture 6-2 - Extension Point and Alternative Flow
3. **Video:** Lecture 6-3 - SubFlow
4. **Reading:** Lecture 6 - System Requirements Capture
5. **Reading:** Lecture 6 - Use Case Detailed Specification Example
6. **Reading:** Lecture 6 - Use Case Model Exercise

Week 7 :Non-Functional Requirements

1. [Video: Lecture 7-1 - Non-Functional Requirements](#)
2. **Video:** Lecture 7-2 - Validate System Requirements

3. **Reading:** Lecture 7 - System Requirements Capture
4. **Reading:** Lecture 7 - Use Case Modeling Exercise
5. **Reading:** Lecture 7 - System Requirements Capture Exercise

Graded: Quiz 7 - System Requirements Capture

Course 2 – Software Engineering:Implementation and Testing

Week 1 : Introduction to Software Development

1. [Video: Lecture 1-1 Introduction to Software Development](#)
2. **Video:** Lecture 1-2 Project Risks
3. **Video:** Lecture 1-3 Project Planning
4. **Reading:** Lecture 1 - Software Development

Graded: Quiz 1 - Software Development

Week 2 : Software Development Processes

1. [Video: Lecture 2-1 Software Development Processes](#)
2. **Video:** Lecture 2-2 Agile
3. **Video:** Lecture 2-3 Unified Process
4. **Reading:** Lecture 2 - Software Development
5. **Reading:** Lecture 2 - Software Development Case Study Exercise

Graded: Quiz 2 - Software Development

Week 4 :Debugging and Configuration Management

1. [Video: Lecture 4-1 Debugging](#)
2. **Video:** Lecture 4-2 Configuration Management
3. **Reading:** Lecture 4 Implementation

Graded: Quiz 4 - Implementation

Week 5: White Box Testing

1. [Video: Lecture 5-1 Testing](#)
2. **Video:** Lecture 5-2 Design Tests
3. **Video:** Lecture 5-3 Basis Path Testing
4. **Reading:** Lecture 5 - Testing
5. **Reading:** Lecture 5 - Basis Path Testing Exercise

Graded: Quiz 5 - Testing

Week 6 :Black Box Testing

1. [Video: Lecture 6-1 White Box Testing](#)
2. **Video:** Lecture 6-2 Black Box Testing
3. **Video:** Lecture 6-3 Regression Testing
4. **Reading:** Lecture 6 - Testing
5. **Reading:** Lecture 6 - Black Box Testing Exercise

Graded: Quiz 6 - Testing

Week 7: Acceptance Testing

1. [Video: Lecture 7-1 Perform Tests](#)
2. **Video:** Lecture 7-2 Unit/Integration/System Testing
3. **Video:** Lecture 7-3 Acceptance Testing Example
4. **Reading:** Lecture 7 - Testing
5. **Reading:** Lecture 7 - Acceptance Testing Exercise

Graded: Quiz 7 - Testing

Course 3 - Software Engineering: Software Design and Project Management

Week 1 : System Analysis and Design

1. [Video: Lecture 1-1 Introduction to System Design and Analysis](#)
2. **Video:** Lecture 1-2 Architectural Design and Analysis
3. **Video:** Lecture 1-3 Use Case Analysis
4. **Video:** Lecture 1-4 Class Design
5. **Reading:** Lecture 1 - System Design and Analysis
6. **Reading:** Lecture 1 - System Design Optimization Exercise

Graded: Quiz 1 - System Analysis and Design

Week 2 :Design Patterns

1. [Video: Lecture 3-1 Design Pattern](#)
2. **Video:** Lecture 3-2 Strategy Pattern
3. **Video:** Lecture 3-3 Observer Pattern
4. **Reading:** Lecture 3 - System Analysis and Design
5. **Reading:** Lecture 3 - Design Pattern Exercise

Graded: Quiz 3 - System Analysis and Design

Week 3 : Software Quality Assurance

1. [Video: Lecture 5-1 Achieving Software Quality](#)
2. **Video:** Lecture 5-2 Software Quality Assurance Activities
3. **Video:** Lecture 5-3 Achieving Product Quality
4. **Video:** Lecture 5-4 Achieving Project-Process-People Quality
5. **Reading:** Lecture 5 - Software Quality Assurance

Graded: Quiz 5 - Software Quality Assurance

Managing Software Development

1. [Video: Lecture 6-1 Project Management](#)
2. **Video:** Lecture 6-2 Software Development Plan
3. **Video:** Lecture 6-3 Project Tracking and Control
4. **Reading:** Lecture 6 - Managing Software Development

Graded: Quiz 6 - Managing Software Development