1. **Course number and name:** CMSE 312 Software Engineering
2. **Credits and contact hours, and categorization:** Credits: 4, Contact hours: 6, Engineering Sciences and Design
3. **Course Instructor:** Prof. Dr. Duygu Çelik Ertuğrul
4. **Textbook:**
* *Ian Sommerville, Engineering Software Products: An Introduction to Modern Software Engineering, ISBN-10: 013521064X • ISBN-13: 9780135210642 ©2020 • Pearson • Paper, 352 pp, Published 18 Feb 2019*
* *Ian Sommerville, Software Engineering 10e, Global Edition, 2016.*

**Other supplemental materials:**

* *Stephen Schach, Object-Oriented and Classical Software Engineering. 7th ed., McGraw-Hill, 2007.*
* *Shari Lawrence Pfleeger, Joanne M. Atlee, Software Engineering: Theory and Practice, 4/E ISBN-10: 0136061699 • ISBN-13: 9780136061694 ©2010 • Pearson • Cloth, 800 pp, Modern Systems Analysis and Design, 8th Edition, Valacich & George ©2017 | Adobe Reader | ISBN-13: 9780134205663,* [*https://www.vitalsource.com/educators/textbooks?term=9780134205663*](https://www.vitalsource.com/educators/textbooks?term=9780134205663)
* *Project Management: Process, Technology and Practice, Ganesh Vaidyanathan, Indiana University, South Bend, ISBN-10: 0132807181 • ISBN-13: 9780132807180 ©2013.Modern Systems Analysis and Design, 8th Edition, Valacich & George ©2017 | Adobe Reader | ISBN-13: 9780134205663.*
1. **Specific course information**
2. **Catalog description:** This course provides a comprehensive introduction to the principles of software engineering, emphasizing the key phases of the software development process, including project planning, feasibility studies, requirements analysis, specification, design, implementation, testing, quality assurance, documentation, and maintenance. Students will explore various software development methodologies such as Waterfall, V-Model, Spiral, RAD, and Agile, while gaining both theoretical and practical skills in requirements engineering, focusing on gathering, specifying, validating, and evolving requirements. The curriculum covers critical technical document preparation, including Software Requirements Specification (SRS) and Software Design Specification (SDS), as well as software architecture models and modular system design. In the verification and validation section, students will engage in practical work related to various testing techniques, including unit testing, integration testing, system testing, and User Acceptance Testing (UAT), conducting validation studies on test planning, quality assurance strategies, and debugging techniques. The course also addresses project management topics, covering essential elements such as project scope, time and cost estimation processes, risk analysis, and quality management, equipping students with the knowledge and skills necessary to apply estimation methods like COCOMO and utilizing project time management tools such as Gantt analysis, Program Evaluation Review Technique (PERT), and Critical Path Method (CPM). To enhance their communication skills, students will prepare structured written reports and deliver oral presentations, gain practical experience in developing software solutions and understanding the societal and global impacts of software engineering through term application projects. In summary, this course blends theoretical knowledge with practical applications to effectively prepare students for managing software development projects and contributing to successful project outcomes.
3. **Prerequisite:** CMPE211 - Object Oriented Programming
4. **Required/elective/selected elective:** Required
5. **Specific goals for the course**
6. **Course outcomes:** On successful completion of the course, students will be able to:
7. Learn the Software Design Project Process Phases
8. Learn the Project Management Process
9. Learn Software Process Estimation Techniques
10. Learn UML tools for Project Analysis and Requirements Specification
11. Learn Analysis of Large Problems
12. Learn Task Distribution, Scheduling and Assignment of tasks to team members
13. Learn Software Design Techniques, modular design
14. Learn Software Testing Principles
15. Learn 3-4 Specific Methods in Software Testing
16. Design a Software System for Term Project
17. Write Progress, Intermediate and Final Reports for the Project
18. Make Presentations for Project Progress, and for Final Project status with demo
19. **Student outcomes listed in Criterion 3**
20. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
21. an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors
22. an ability to communicate effectively with a range of audiences
23. an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts
24. an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives
25. an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions
26. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies.
27. **Topics covered**
* **Foundations for Systems Development**
	+ Chapter 1: Introduction
	+ Chapter 2: SDLC Models
	+ Chapter 3: Software (SW) Processes
	+ Chapter 4: Coping with Changes
	+ Chapter 5: Software (SW) Process Improvement
* **Planning**
	+ Chapter 6: Project Metrics - COCOMO
	+ Chapter 7: Time Management - CPM, PERT, Crashing
	+ Chapter 8: Project Management (PM) Tools
* **Analysis**
	+ Chapter 9: Requirements Engineering (Req Eng)
	+ Chapter 10: Risk Management
* **Design**
	+ Chapter 11: Design - DFD
	+ Chapter 12: UML Design - Use Case Diagram
	+ Chapter 13: UML Design - Activity Diagram
	+ Chapter 14: UML Design - Sequence Diagram
	+ Chapter 15: UML Design - BPMN Diagram
	+ Chapter 16: UML Design - Class Diagram
	+ Chapter 17: UML Design – Design Database, ER Diagram
* **Implementation and Test**
	+ Chapter 18: Testing
	+ Chapter 19: Quality-and-Performance