1. **Course number and name:** CMSE 201 Fundamentals of Software Engineering
2. **Credits and contact hours, and categorization:** Credits: 4, Contact hours: 6, Engineering Sciences and Design
3. **Course Instructor:** Prof. Dr. Alexander Chefranov
4. **Textbook:**

* *Ian Sommerville, Engineering Software Products: An Introduction to Modern Software Engineering, ISBN-10: 013521064X • ISBN-13: 9780135210642 ©2020, Pearson, 352 pp, Published 18 Feb 2019*
* *Ian Sommerville, Software Engineering 10e, Global Edition, 2016.*

**Other supplemental materials:**

* *Shari Lawrence Pfleeger, Joanne M. Atlee, Software Engineering: Theory and Practice, 4/E ISBN-10: 0136061699 • ISBN-13: 9780136061694 ©2010, Prentice Hall, 756 pp,*
* Joseph S. Valacich and Joey F. George, *Modern Systems Analysis and Design, 9th Edition, Pearson Education Limited ©2021* **ISBN 10:** 1-292-35162-4, **ISBN 13:** 978-1-292-35162-9
* *Ganesh Vaidyanathan, Project Management: Process, Technology and Practice, Indiana University, South Bend, ISBN-10: 0132807181 , ISBN-13: 9780132807180 ©2013.*
* *Sommerville, I., Software Engineering, 8th Ed., ISBN 7-111-19770-4, 2006*
* *Pressman, R.S., Software Engineering: A Practitioner Approach, 5th Ed., ISBN -07-365578-3, 2001*

1. **Specific course information**
2. **Catalog description:** Phases of the software development process. Estimation, system modeling, requirements analysis. Project management, modular software design, object-oriented analysis, and design techniques. Design documentation using symbolic representations, UML diagrams. Software testing, quality issues.
3. **Prerequisite:** None
4. **Required/elective/selected elective:** Required
5. **Specific goals for the course**
6. **Course outcomes:** After successfully completing this course, students will be able to:
7. identify, formulate, and solve software engineering problems, including the specification, design, implementation, and testing of software systems that meet specification, performance, maintenance, and quality requirements.
8. elicit, analyze and specify software requirements through a productive working relationship with various stakeholders of a software development project.
9. work effectively as a team member.
10. understand professional, ethical, and social responsibility of a software engineer.
11. participate in design, development, deployment, and maintenance of a small-to-medium scale software development project.
12. convey technical material through oral presentation and interaction with an audience.
13. convey technical material through written reports which satisfy accepted standards for writing style.
14. use Unified Modeling Language in software specification documents.
15. evaluate the impact of potential solutions to software engineering problems in a global society, using the knowledge of contemporary issues and emerging software engineering trends, models, tools, and techniques.
16. apply modular software design, object-oriented analysis and design techniques, and construction.
17. apply the cost and time estimations of a software development project management.
18. apply computer science theory and software engineering fundamentals to produce software solutions.
19. understand the methods of behavioral and static modelling of a software system.
20. understand notions of software quality assurance and control activities.
21. **Student outcomes listed in Criterion 3**
22. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
23. 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
24. an ability to communicate effectively with a range of audiences
25. 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
26. 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
27. an ability to develop and conduct appropriate experimentation, analyze, and interpret data, and use engineering judgment to draw conclusions
28. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies
29. **Topics covered**

* **Software Engineering and Processes**: Introduction to Software Engineering, Software Evolution, Software Characteristics, Software Crisis: Problem and Causes, Software development life-cycle, Software process models (Waterfall, Incremental, and Evolutionary process models and Agile), Software quality concepts, process improvement, software process, capability maturity models, Personal Software process and Team Software Process, Overview of Agile Process.
* **Requirements Engineering**: Problem Analysis, Requirement elicitation and Validation, Requirements modeling: Scenarios, Information and analysis classes, flow and behavioral modeling, documenting Software Requirement Specification (SRS).
* **Software Design and construction**: System design principles: levels of abstraction (architectural and detailed design), separation of concerns, information hiding, Structured design (top-down functional decomposition), object oriented design, event driven design, component-level design, test driven design, design patterns.
* **Software Verification and Validation**: Levels of Testing, Functional Testing, Structural Testing, Test Plan, Test Case Specification, Software Testing Strategies, Verification & Validation, Unit, Integration Testing, Top Down and Bottom Up Integration Testing, Alpha & Beta Testing, White box and black box testing techniques, System Testing and Debugging.
* **Software Project Management**: Software project estimation of scope (LOC, FP etc), time (PERT/CPM Networks), and cost (COCOMO models), quality management, project plan for software quality control and assurance, earned value analysis.
* **Advanced Topics**: Formal specification, CASE Tools, Software Business Process Reengineering, Configuration Management.