SOFTWARE ENGINEERING (SE)

Courses primarily for undergraduates:

SE 1010: Software Engineering Orientation
Credits: Required. Contact Hours: Lecture 1.
Introduction to the procedures, policies, and resources of Iowa State University and the Software Engineering Program. Offered on a satisfactory-fail basis only.

SE 1660: Careers in Software Engineering
Credits: Required. Contact Hours: Lecture 1.
Overview of the nature and scope of the software engineering profession, relationship of coursework to careers, and program of study planning. Offered on a satisfactory-fail basis only.

SE 1850: Problem Solving in Software Engineering
Credits: 3. Contact Hours: Lecture 2, Laboratory 2.
Prereq: Credit or concurrent enrollment in MATH 1430 *(or satisfactory scores on mathematics placement examinations)*
Introduction to software engineering and computer programming. Systematic thinking process for problem solving in the context of software engineering. Group problem solving. Solving software engineering problems and presenting solutions through computer programs, written documents and oral presentations. Introduction to principles of programming, software design, and extensive practice in design, writing, running, debugging, and reasoning about programs. Satisfactory placement scores can be found at: https://math.iastate.edu/academics/undergraduate/aleks/placement/. Graduation Restriction: Only one of ENGR 1600, ABE 1600, AERE 1600, BME 1600, CE 1600, CHE 1600, CPRE 1850, EE 1850, IE 1480, ME 1600, and SE 1850 may count towards graduation.

SE 1860: Problem Solving in Software Engineering II
Credits: 1. Contact Hours: Laboratory 2.
Prereq: SE 1850
Group projects in software engineering. Work effectively in teams to solve problems and provide technical reports and presentations. Self-directed team based projects that are representative of problems faced by software engineers. (Typically Offered: Spring)

SE 3090: Software Development Practices
(Cross-listed with COMS 3090).
Credits: 3. Contact Hours: Lecture 3, Discussion 1.
*Prereq: Minimum of C- in (*COMS 2280*;* MATH 1650*)*
Practical introduction to methods for managing software development. Software engineering concepts, practices and tools. Requirements analysis, structured and object-oriented design, coding, testing, and maintenance. Software process models, software tools and environments. Programming projects that provide exposure to information management techniques, client/server model, networking and communication. (Typically Offered: Fall, Spring)

SE 3170: Introduction to Software Testing
Credits: 3. Contact Hours: Lecture 3.
*Prereq: (*COMS 2300 *or* CPRE 3100*); (*COMS 3090 *or SE 3090)*
Basic principles and techniques for software testing. Test requirements and management. Test design techniques, evaluation metrics, model-based testing, unit testing, system and integration testing. Software testing tools and programming projects.

SE 3190: Construction of User Interfaces
(Cross-listed with COMS 3190).
Credits: 3. Contact Hours: Lecture 3.
Prereq: COMS 2280
Overview of user interface design. Evaluation and testing of user interfaces. Review of principles of object orientation, object oriented design and analysis using UML in the context of user interface design. Design of windows, menus and commands. Developing Web and Windows-based user-interfaces. Event-driven programming. Introduction to Frameworks and APIs for the construction of user interfaces. (Typically Offered: Fall, Spring)

SE 3290: Software Project Management
(Cross-listed with CPRE 3290).
Credits: 3. Contact Hours: Lecture 3.
Prereq: COMS 3090
Process-based software development. Capability Maturity Model (CMM). Project planning, cost estimation, and scheduling. Project management tools. Factors influencing productivity and success. Productivity metrics. Analysis of options and risks. Version control and configuration management. Inspections and reviews. Managing the testing process. Software quality metrics. Modern software engineering techniques and practices.

SE 3390: Software Architecture and Design
(Cross-listed with CPRE 3390).
Credits: 3. Contact Hours: Lecture 3.
Prereq: SE 3190
Modeling and design of software at the architectural level. Architectural styles. Basics of model-driven architecture. Object-oriented design and analysis. Iterative development and unified process. Design patterns. Design by contract. Component based design. Product families. Measurement theory and appropriate use of metrics in design. Designing for qualities such as performance, safety, security, reliability, reusability, etc. Analysis and evaluation of software architectures. Introduction to architecture definition languages. Basics of software evolution, reengineering, and reverse engineering. Case studies. Introduction to distributed system software.

SE 3420: Principles of Programming Languages
(Cross-listed with COMS 3420).
Credits: 3. Contact Hours: Lecture 3, Discussion 1.
Prereq: Minimum of C- in COMS 2280 *and* MATH 1650*;* COMS 2300 *or* CPRE 3100
Study of concepts in programming languages, especially functional programming concepts. Overview of major programming paradigms, their relationship, and tradeoffs among paradigms enabling sound choices of programming language for application-specific development. Programming projects. (Typically Offered: Fall, Spring)

SE 3620: Object-Oriented Analysis and Design
(Cross-listed with COMS 3620).
Credits: 3. Contact Hours: Lecture 3.
Prereq: ENGL 2500 *and Minimum of C- in* COMS 2280 *and* MATH 1650
Object-oriented requirements analysis and systems design. Analysis and design methodologies including use case and Unified Modeling Language (UML). Design principles, heuristics, and patterns. Architectural patterns and alternative programming paradigms. Group design and programming project. (Typically Offered: Fall, Spring)

SE 4090: Software Requirements Engineering
(Cross-listed with COMS 4090).
Credits: 3. Contact Hours: Lecture 3.
Prereq: COMS 3090
The requirements engineering process including elicitation, requirements analysis fundamentals, requirements specification and communication, and requirements evaluation. Modeling of functional requirements and nonfunctional requirements, traceability, and requirements change management. Case studies and software projects.

SE 4120: Formal Methods in Software Engineering
(Cross-listed with CPRE 4120/ COMS 4120).
Credits: 3. Contact Hours: Lecture 3.
Prereq: COMS 3110*; (*STAT 3050 *or* STAT 3300 *or* STAT 3410*)*
A study of formal techniques for model-based specification and verification of software systems. Topics include logics, formalisms, graph theory, numerical computations, algorithmsand tools for automatic analysis of systems. Graduate credit requires in–depth study of concepts.

SE 4130: Program Analysis
(Cross-listed with COMS 4130).
Credits: 3. Contact Hours: Lecture 3.
*Prereq: (*COMS 3270 *or* CPRE 2880*);* COMS 3420
Algorithms, AI techniques and tools for automatically reasoning about code and program executions. Theory and foundations related to control flow analysis, dataflow analysis, abstract interpretation, and symbolic execution. Applications of program analysis to bug detection, test input generation, debugging, program repair, specification inference and trustworthy AI engineering. Concepts, algorithms, tools, benchmarks, methodologies for solving problems using program analysis and for preparing research in program analysis.

SE 4160: Software Evolution and Maintenance
(Cross-listed with CPRE 4160).
Credits: 3. Contact Hours: Lecture 3.
Prereq: COMS 3090
Practical importance of software evolution and maintenance, systematic defect analysis and debugging techniques, tracing and understanding large software, impact analysis, program migration and transformation, refactoring, tools for software evolution and maintenance, experimental studies and quantitative measurements of software evolution. Written reports and oral presentation.

SE 4170: Software Testing
(Cross-listed with COMS 4170).
Credits: 3. Contact Hours: Lecture 3.
Prereq: COMS 3090*; (*COMS 2300 *or* CPRE 3100*);* ENGL 2500
An introduction to software testing principles and techniques. Test models, test design, test adequacy criteria; regression, integration, and system testing; and software testing tools.

SE 4190: Software Tools for Large Scale Data Analysis
(Cross-listed with CPRE 4190).
Credits: 4. Contact Hours: Lecture 3, Laboratory 3.
Prereq: COMS 3630 *or* COMS 3520 *or* CPRE 3080*;* COMS 2280
Software tools for managing and manipulating large volumes of data, external memory processing, large scale parallelism, and stream processing, data interchange formats. Weekly programming labs that involve the use of a parallel computing cluster.

SE 4210: Software Analysis and Verification for Safety and Security
(Cross-listed with CPRE 4210).
Credits: 3. Contact Hours: Lecture 3.
Significance of software safety and security; various facets of security in cyber-physical and computer systems; threat modeling for software safety and security; and categorization of software vulnerabilities. Software analysis and verification: mathematical foundations, data structures and algorithms, program comprehension, analysis, and verification tools; automated vs. human-on-the-loop approach to analysis and verification; and practical considerations of efficiency, accuracy, robustness, and scalability of analysis and verification. Cases studies with application and systems software; evolving landscape of software security threats and mitigation techniques. Understanding large software, implementing software analysis and verification algorithms. (Typically Offered: Fall, Spring)

SE 4220: Cloud Computing - Software Development
Credits: 3. Contact Hours: Lecture 3.
*Prereq: (*SE 3090 *or SE 3390); (*CPRE 3810 *or* COMS 3210*)*
A comprehensive view of cloud computing with respect to software development from platforms and services to programming and infrastructure. Virtualization and containerization; cloud computing platforms, with examples from currently available cloud services; cloud services for data analytics, machine learning, and devops; programming frameworks for parallel computing in the cloud; distributed storage in the cloud; Container management. Includes homeworks and programming assignments. The programming assignments will be done in AWS. (Typically Offered: Fall)

SE 4400: Principles and Practice of Compiling
(Cross-listed with COMS 4400).
Credits: 3. Contact Hours: Lecture 3.
Prereq: COMS 3310 *or* COMS 3420 *and* COMS 3090 *and* ENGL 2500
Theory of compiling and implementation issues of programming languages. Programming projects leading to the construction of a compiler. Projects with different difficulty levels will be given for 4400 and 5400. Topics include: lexical, syntactic and semantic analyses, syntax-directed translation, code generation, runtime environment and library support.

SE 4900: Independent Study
Credits: 1-30. Repeatable.
Prereq: Senior classification in software engineering; Permission of Instructor

Investigation of an approved topic.

SE 4910: Senior Design Project I and Professionalism
(Cross-listed with CPRE 4910/ EE 4910).
Credits: 3. Contact Hours: Lecture 2, Laboratory 3.
Prereq: CPRE or CYBE major: CPRE 2320*, cr or enrollment in* CPRE 3080*,* ENGL 3140 *or* ENGL 3140H*. EE major:* EE 2320*, cr or enrollment in EE 3220,* ENGL 3140 *or* ENGL 3140H*. SE major:* SE 3170 *and* SE 3390*,* ENGL 3090 *or* ENGL 3140 *or* ENGL 3140H*. CYBE major: Co-req:* CPRE 3080 *or* COMS 3520

Preparing for entry to the workplace. Selected professional topics. Use of technical writing skills in developing project plan and design report; design review presentation. First of two-semester team-oriented, project design and implementation experience. (Typically Offered: Fall, Spring)

SE 4920: Senior Design Project II
(Cross-listed with EE 4920/ CPRE 4920).
Credits: 2. Contact Hours: Lecture 1, Laboratory 3.
Prereq: CPRE 4910 *or EE 4910*
Second semester of a team design project experience. Emphasis on the successful implementation and demonstration of the design completed in EE 4910, CPRE 4910, or SE 4910 and the evaluation of project results. Technical writing of fil project report; oral presentation of project achievements; project poster. (Typically Offered: Fall, Spring)