

Software Engineering Course Descriptions

ECS 1200 Introduction to Engineering and Computer Science (2 semester hours)

Introduction to the Engineering and Computing professions, professional ethics. Overview of ECS curricula, connections among ECS fields and to the basics of sciences, other fields. Basic study, problem solving and other skills needed to succeed as an ECS major. Engineering design and quantitative methods using MATLAB. Multi-disciplinary team projects designed to replicate decision processes in real-world situations. (1-2) Y

SE 2V95 Individual Instruction in Computer Science/Software Engineering (*1-6 semester hours*) Individual study under a faculty member's direction. May be repeated for credit as topics vary (*6 hours maximum*). Consent of instructor required. (Same as [CS 2V95](#)) ([1-6]-0) R

SE 3195 Special Topics in Computer Science/Software Engineering (*1 semester hour*) May be repeated for credit as topics vary (*4 hours maximum*). Must be taken credit/no credit. Consent of instructor required. (Same as [CS 3195](#)) (1-0) R

SE 3306 Mathematical Foundations of Software Engineering (*3 semester hours*) Boolean logic, first-order logic, models of first-order logic. Introduction to program verification, applications in Software Engineering. Completeness Theorem. Regular expressions, regular sets, finite-state machines, and applications in Software Engineering. Graph Theory, graph algorithms. Statecharts, Petri Nets and their role in Software Engineering. Prerequisite: [CS 2305](#) or equivalent. (3-0) S

SE 3340 Computer Architecture (*3 semester hours*) This course introduces the concepts of computer architecture by going through multiple levels of abstraction, and the numbering systems and their basic computations. It focuses on the instruction-set architecture of the MIPS machine, including MIPS assembly programming, translation between MIPS and C, and between MIPS and machine code. General topics include performance calculation, processor datapath, pipelining, and memory hierarchy. Students who have already completed [CS 2310](#) or [CS/SE 4340](#) cannot receive credit for this course. Students cannot get credit for both [CS/SE 3340](#) and [CE/EE 4304](#). Prerequisite: [CS 1337](#) or equivalent. (Same as [CS 3340](#)) (3-0) S

SE 3341 Probability and Statistics in Computer Science and Software Engineering (*3 semester hours*) Axiomatic probability theory, independence, conditional probability. Discrete and continuous random variables, special distributions of importance to CS/SE, and expectation. Simulation of random variables and Monte Carlo methods. Central limit theorem. Basic statistical inference, parameter estimation, hypothesis testing, and linear regression. Introduction to stochastic processes. Illustrative examples and simulation exercises from queuing, reliability, and other CS/SE applications. Students cannot get credit for both [CS/SE 3341](#) and [ENGR 3341](#). Prerequisites: [MATH 1326](#), [MATH 2414](#), or [MATH 2419](#), and [CS 2305](#) or equivalent. (Same as [CS 3341](#)) (3-0) S

CS 3345 Data Structures and Introduction to Algorithmic Analysis (*3 semester hours*) Analysis of algorithms including time complexity and Big-O notation. Analysis of stacks,

queues, and trees, including B-trees. Heaps, hashing, and advanced sorting techniques. Disjoint sets and graphs. Course emphasizes design and implementation. Students cannot get credit for both ECS3345 and CE/TE 3346. Prerequisites: [CS 2305](#) and [CS 2336](#) Prerequisite or corequisite: [CS/SE 3341](#) . (3-0) S

[CS 3354 Software Engineering](#) (*3 semester hours*) Introduction to software life cycle models. Software requirements engineering, formal specification and validation. Techniques for software design and testing. Cost estimation models. Issues in software quality assurance and software maintenance. Prerequisites: [CS 2336](#) or [CS 3333](#), and [CS 2305](#) or equivalent. Pre- or corequisite: [CS 3390](#). (3-0) S

[SE 3376 C/C++ Programming in a UNIX Environment](#) (*3 semester hours*) Advanced programming techniques utilizing procedural and object oriented programming in a UNIX environment. Topics include file input and output, implementation of strings, stacks, queues, lists, and trees, and dynamic memory allocation/management. Design and implementation of a comprehensive programming project is required. Prerequisite: [CS 2336](#) or equivalent. (Same as [CS 3376](#)) (3-0) S

[SE 3V95 Undergraduate Topics in Computer Science/Software Engineering](#) (*2-9 semester hours*) Subject matter will vary from semester to semester. May be repeated for credit as topics vary (*9 hours maximum*). (Same as [CS 3V95](#)) ([2-9]-0) S

[SE 4347 Database Systems](#) (*3 semester hours*) This course emphasizes the concepts and structures necessary for the design and implementation of database management systems. Topics include data models, data normalization, data description languages, query facilities, file organization, index organization, file security, data integrity, and reliability. Prerequisite: [CS 3345](#). (Same as [CS 4347](#)) (3-0) Y

[CS 4348 Operating Systems Concepts](#) (*3 semester hours*) An introduction to fundamental concepts in operating systems: their design, implementation, and usage. Topics include process management, main memory management, virtual memory, I/O and device drivers, file systems, secondary storage management, and an introduction to critical sections and deadlocks. Prerequisites: [CS/SE 3340](#) or equivalent, [CS 3345](#) , and a working knowledge of C and UNIX. (3-0) S

[SE 4351 Requirements Engineering](#) (*3 semester hours*) Introduction to system and software requirements engineering. The requirements engineering process, including requirements elicitation, specification, and validation. Essential words and types of requirements. Structural, informational, and behavioral requirements. Non-functional requirements. Scenario analysis. Conventional, object-oriented and goal-oriented methodologies. Prerequisites: [SE 3306](#), [CS 3354](#) or consent of instructor. (3-0) S

[SE 4352 Software Architecture and Design](#) (*3 semester hours*) Introduction to software design with emphasis on architectural design. Models of software architecture. Architecture styles and patterns, including explicit, event-driven, client-server, and middleware architectures. Decomposition and composition of architectural components and interactions. Use of non-

functional requirements for tradeoff analysis. Component based software development, deployment and management. Prerequisites: [SE 3306](#), [CS 3354](#) or consent of instructor. (3-0) S

[SE 4367](#) Software Testing, Verification, Validation and Quality Assurance (*3 semester hours*). Methods for evaluating software for correctness, and reliability including code inspections, program proofs and testing methodologies. Formal and informal proofs of correctness. Code inspections and their role in software verification. Unit and system testing techniques, testing tools and limitations of testing. Statistical testing, reliability models. Prerequisites: [SE 3306](#), [CS 3354](#) or consent of instructor. (3-0) S

[SE 4376](#) Object-Oriented Programming Systems (*3 semester hours*) In-depth study of the features/advantages of object-oriented approach to problem solving. Special emphasis on issues of object-oriented analysis, design, implementation, and testing. Review of basic concepts of object-oriented technology (abstraction, inheritance, and polymorphism). Object-oriented programming languages, databases, and productivity tools. Prerequisite: [CS 2336](#) or equivalent. (Same as [CS 4376](#)) (3-0) S

[SE 4381](#) Software Project Planning and Management (*3 semester hours*) Planning and managing of software development projects. Software process models, ISO 9000, SEI's Capability Maturity Model, continuous process improvement. Planning, scheduling, tracking, cost estimation, risk management, configuration management. Prerequisite: [CS 3354](#). (3-0) Y

[SE 4399](#) Senior Honors in Computer Science/Software Engineering (*3 semester hours*) For students conducting independent research for honors theses or projects. (Same as [CS 4399](#)) (3-0) R

[SE 4485](#) Software Engineering Project (*4 semester hours*) This course is intended to complement the theory and to provide an in-depth, hands-on experience in all aspects of software engineering. The students will work in teams on projects of interest to industry and will be involved in analysis of requirements, architecture and design, implementation, testing and validation, project management, software process, software maintenance, and software re-engineering. Prerequisites: at least two of [SE 4351](#), [SE 4352](#), [SE 4367](#), [SE 4381](#). (4-1) S

[SE 4V95](#) Undergraduate Topics in Computer Science/Software Engineering (*1-9 semester hours*) Subject matter will vary from semester to semester. May be used as SE Guided Elective on SE degree plans. May be repeated for credit as topics vary (*9 hours maximum*). (Same as [CS 4V95](#)) ([1-9]-0) R

[SE 4V98](#) Undergraduate Research in Computer Science/Software Engineering (*1-9 semester hours*) Topics will vary from semester to semester. May be repeated for credit (*9 hours maximum*). Consent of instructor required. (Same as [CS 4V98](#)) ([1-9]-0) R