SOFTWARE ENGINEERING

For the undergraduate curriculum in Software Engineering (http://www.se.iastate.edu) leading to the degree Bachelor of Science. The Software Engineering Program is accredited by the Engineering Accreditation Commission of ABET, http://www.abet.org.

The Bachelor of Science degree in software engineering is jointly administered by the College of Engineering and the College of Liberal Arts and Sciences. The Software Engineering program provides undergraduate students with the opportunity to learn software engineering fundamentals, to study applications of state-of-the art software technologies and to prepare for the practice of software engineering. The student-faculty interaction necessary to realize this opportunity occurs within an environment motivated by the principle that excellence in undergraduate education is enhanced by an integrated commitment to successful, long-term research and outreach programs.

The software engineering curriculum offers emphasis areas in software engineering principles, process and practice. Students may also take elective courses in computer engineering and computer science.

Program Educational Objectives

Within five years of graduation, the graduates should:

1. attain a **productive career** in Software Engineering or related fields;

2. attain **leadership** roles and become **effective collaborators** to advance professional and organizational goals;

3. engage in **lifelong learning** and professional development.

4. encourage and support **diversity** and **inclusiveness** in their workplace

We expect that these objectives will be manifested in our graduates through the following five key attributes: (a) *peer-recognized expertise*, (b) *engagement in professional practice*, (c) *sustained learning*, (d) *leadership* and (e) *teamwork*.

Demonstration of expertise involves applying state-of-the-art practices for solving problems in the design, development, validation, evolution and sustainment of (software) products. Demonstration of professional engagement involves contributing locally and globally to the use of ethical, competent, and creative practices in industry, academia or the public sector. Demonstration of sustained learning involves the ability to adapt to rapid technological, environmental, and organizational changes through self-study and group study and through opportunities of professional development or graduate study. Demonstration of leadership involves the ability to take initiative, and to facilitate the advancements of individuals and the community by influencing others and by having a widespread, positive impact on critical issues and projects. Finally, demonstration of teamwork involves the ability to work with collaborators who have varied expertise, and with diverse cultural and interdisciplinary backgrounds.

As a complement to the instructional activity, the College of Engineering and the College of Liberal Arts and Sciences provide opportunities for each student to have experience with broadening activities. Students have the opportunity to gain practical industry experience in the cooperative education and internship program. Students have the opportunity to participate in advanced research activities. Through international exchange programs, students learn about engineering practices in other parts of the world.

Curriculum in Software Engineering

The Software Engineering program is an interdisciplinary program delivered jointly by the College of Engineering and the College of Liberal Arts & Sciences.

Leading to the degree bachelor of science.

Total credits required: 125 cr. Any transfer credit courses applied to the degree program require a grade of C or better (but will not be calculated into the ISU cumulative GPA, Basic Program GPA or Core GPA). See also Basic Program and Special Programs. Note: Pass/Not Pass credits cannot be used to meet graduation requirements.
International Perspectives: 3 cr. [1]
U.S. Diversity: 3 cr. [1]
Communication Proficiency/Library requirement:

ENGL 150	Critical Thinking and Communication (Must have a C or better in this course)	3
ENGL 250	Written, Oral, Visual, and Electronic Composition (Must have a C or better in this course)	3
LIB 160	Information Literacy	1
Choose one of the following:		3
ENGL 309	Proposal and Report Writing (C or better in this course)	
ENGL 314	Technical Communication (C or better in this course)	
Total Credits		10

General Education Electives: 15 cr. [2]

Choose 1 course from the following:		3
ECON 101	Principles of Microeconomics	
ECON 102	Principles of Macroeconomics	
I E 305	Engineering Economic Analysis	
Arts and Humanities		6
Social Sciences		3
Additional Arts and Humanities or Social Sciences course		3
Total Credits		15

Basic Program: 27 cr.

A minimum GPA of 2.00 is required for this set of courses, including any transfer courses (please note that transfer course grades will not be calculated into the Basic Program GPA). See Requirement for Entry into Professional Program in College of Engineering Overview section.

CHEM 167	General Chemistry for Engineering Students	4
or CHEM 177	General Chemistry I	
ENGL 150	Critical Thinking and Communication (Must have a C or better in this course)	3
ENGL 250	Written, Oral, Visual, and Electronic Composition (Must have a C or better in this course)	3
S E 101	Software Engineering Orientation [3]	R
S E 185	Problem Solving in Software Engineering [3]	3
LIB 160	Information Literacy	1
MATH 165	Calculus I	4
MATH 166	Calculus II	4
PHYS 221	Introduction to Classical Physics I	5
Total Credits		27

Math and Physical Science: 11 cr.

COM S 227	Introduction to Object-oriented Programming	4
COM S 228	Introduction to Data Structures	3
MATH 267	Elementary Differential Equations and Laplace Transforms	4
Total Credits		11

Software Engineering Core: 37 cr.

A minimum GPA of 2.00 is required for this set of courses, including any transfer courses (please note that transfer course grades will not be calculated into the Core GPA):

CPR E 281	Digital Logic	4
Choose one of the following:		3
COM S 327	Advanced Programming Techniques	
CPR E 288	Embedded Systems I: Introduction	
Choose one of the following:		3
COM S 321	Introduction to Computer Architecture and Machine-Level Programming	
CPR E 381	Computer Organization and Assembly Level Programming	
Choose one of the following:		3
COM S 352	Introduction to Operating Systems	
CPR E 308	Operating Systems: Principles and Practice	
Choose one of the following:		3
COM S 230	Discrete Computational Structures	
CPR E 310	Theoretical Foundations of Computer Engineering	
COM S 311	Design and Analysis of Algorithms	3
COM S 363	Introduction to Database Management Systems	3
COM S 309	Software Development Practices	3
S E 319	Software Construction and User Interfaces	3
S E 329	Software Project Management	3
S E 339	Software Architecture and Design	3
S E 421	Software Analysis and Verification for Safety and Security	3
Note: CPR E 288, CPR E 381 and CPR E 308 are 4-credit courses. The core credit requirement (37 credits) is given in terms of 3-credit courses. If the 4-credit courses are taken instead, then the extra credits will be used as credits for Supplementary Electives.		
Total Credits		37

Other Remaining Courses: 35 cr.

S E 491	Senior Design Project I and Professionalism	3
S E 492	Senior Design Project II	2
SP CM 212	Fundamentals of Public Speaking	3
STAT 330	Probability and Statistics for Computer Science	3
One of the following ENGL courses (with a C or better in this course)		3
ENGL 309	Proposal and Report Writing	
ENGL 314	Technical Communication	
Math Elective: Choose one from the following list		3
MATH 207	Matrices and Linear Algebra	
MATH 265	Calculus III	
MATH 304	Combinatorics	
MATH 314	Graph Theory	
MATH 317	Theory of Linear Algebra	
Software Engineering Elective [2]		6
Supplementary Elective [2]		9
Open Elective [2]		3
Total Credits		35

Seminar/Co-op/Internships

S E 166	Careers in Software Engineering	R
S E 494	Software Engineering Portfolio Development	R
Co-op or internship (S E 396, S E 397, S E 398) is optional		

Transfer Credit Requirements

The degree program must include a minimum of 30 credits at the 300-level or above in professional and technical courses earned at ISU in order to receive a B.S. in software engineering. These 30 credits must include S E 491 Senior Design Project I and Professionalism and S E 492 Senior Design Project II and credits in the core professional curriculum and/or in technical electives. The software engineering degree program requires a grade of C or better for any transfer credit course that is applied to the degree program.

1. These university requirements will add to the minimum credits of the program unless the university-approved courses are also approved by the department to meet other course requirements within the degree program. U.S Diversity and International Perspectives courses may not be taken Pass/Not Pass.
2. Choose from department approved lists. (http://www.se.iastate.edu/academics)
3. See Basic Program for Professional Engineering Curricula for accepted substitutions for curriculum designated courses in the Basic Program.

See also: A 4-year plan of study grid showing course template by semester.

Note: International perspectives and U.S. diversity courses are used to meet the general education electives (http://www.se.iastate.edu/academics).

Plan of Study - 4 Year Plan [*]

Freshman

Fall	Credits	Spring	Credits
MATH 165	4	COM S 227	4
ENGL 150	3	MATH 166	4
S E 101	0	S E 166	0
LIB 160	1	PHYS 221	5
CHEM 167 or 177	4	Economics Elective	3
S E 185	3		
	15		16

Sophomore

Fall	Credits	Spring	Credits
CPR E 281[1]	4	S E 319[1]	3
ENGL 250	3	COM S 327 or CPR E 288[1, 2]	3
MATH 267	4	Math Elective	3
COM S 228	3	General Education Elective	3
SP CM 212	3		
	17		12

Junior

Fall	Credits	Spring	Credits
COM S 363[1]	3	S E 329[1]	3
COM S 230 or CPR E 310[1]	3	COM S 352 or CPR E 308[1, 2]	3
COM S 309[1]	3	ENGL 314 or 309	3
COM S 321 or CPR E 381[1, 2]	3	COM S 311[1]	3
General Education Elective	3	S E 339[1]	3
		Open Elective	3
	15		18

Senior

Fall	Credits	Spring	Credits
S E 494	0	S E 492	2
S E 491	3	Supplementary Electives	9
STAT 330	3	Software Engineering Electives	3
S E 421	3	General Education Elective	3
General Education Elective	3		
Software Engineering Elective	3		
	15		17

[*] Total credits required - 125 credits. Any transfer credit courses applied to the degree program require a grade of C or better (but will not be calculated into the ISU Cumulative GPA, Basic Program GPA or Core GPA). See also Basic Program and Special Programs.
Basic Program - A minimum GPA of 2.00 is required for this set of courses, including any transfer courses (please note that transfer course grades will not be calculated into the Basic Program GPA)
Must receive a C or better grade in required English classes.
General Educational Elective courses to be selected in consultation with SE advisers from a list of approved courses. They must include courses that satisfy university and college general education requirements. These courses include courses or categories of courses such as international perspectives and U.S. diversity, language, arts and humanities, and social sciences. Pass/Not Pass credit is not accepted.
Software Engineering Electives and Supplementary Electives must be selected from the program-approved list (http://www.se.iastate.edu/academics).

[1] Core Professional Curriculum (A minimum GPA of 2.00 is required for this set of courses, including any transfer courses but transfer course grades will not be calculated into the Core GPA)

[2] Students who take the 4-credit lab courses CPR E 288, CPR E 308, and CPR E 381 instead of the corresponding 3-credit alternatives can apply the additional credits toward Supplementary Electives. The total number of credits required in the Software Engineering Program remains the same for all students.

Courses primarily for undergraduates:

S E 101: Software Engineering Orientation
Cr. R.
Introduction to the procedures, policies, and resources of Iowa State University and the department of Computer Science and Electrical and Computer Engineering. Information on engineering and computer-based professions.

S E 166: Careers in Software Engineering
Cr. R.
Overview of the nature and scope of the software engineering profession. Relationship of coursework to careers. Departmental rules, student services operations, degree requirements, program of study planning, career options, and student organizations.

S E 185: Problem Solving in Software Engineering
(2-2) Cr. 3.
Prereq: MATH 143 or satisfactory scores on mathematics placement examinations; credit or enrollment in MATH 165
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.

S E 319: Software Construction and User Interfaces
(Cross-listed with COM S). (3-0) Cr. 3.
Prereq: COM S 228
Basic theory of grammars, parsing. Language paradigms. State transition and table-based software design. Review of principles of object orientation, object oriented analysis using UML. Frameworks and APIs. User interface architecture, evaluation of user interface. Design of windows, menus, and commands. Introduction to formal specification and model-based software design. Introduction to domain-specific software engineering.

S E 329: Software Project Management
(Cross-listed with CPR E). (3-0) Cr. 3.
Prereq: COM S 309
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.

S E 339: Software Architecture and Design
(Cross-listed with CPR E). (3-0) Cr. 3.
Prereq: S E 319
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.

S E 342: Principles of Programming Languages
(Cross-listed with COM S). (3-1) Cr. 3.
Prereq: Minimum of C- in COM S 228 and MATH 165; COM S 230 or CPR E 310
Study of concepts in programming languages and major programming paradigms, especially functional programming. Special emphasis on design tradeoffs that enable students to make sound choices of programming languages for a given software development task. Programming projects.

S E 396: Summer Internship
Cr. R. Repeatable. SS.
Prereq: Permission of department and Engineering Career Services
Professional work period of at least 10 weeks during the summer. Students must register for this course prior to commencing work. Offered on a satisfactory-fail basis only.

S E 398: Cooperative Education (Co-op)
Cr. R. Repeatable. F.S.
Prereq: Permission of department and Engineering Career Services
Professional work period. One semester per academic or calendar year. Students must register for this course before commencing work. Offered on a satisfactory-fail basis only.

S E 409: Software Requirements Engineering
(3-0) Cr. 3.
Prereq: COM S 309; for graduate credit: graduate standing or permission of instructor
The requirements engineering process including identification of stakeholders requirements elicitation techniques such as interviews and prototyping, analysis fundamentals, requirements specification, and validation. Use of Models: State-oriented, Function-oriented, and Object-oriented. Documentation for Software Requirements. Informal, semi-formal, and formal representations. Structural, informational, and behavioral requirements. Non-functional requirements. Use of requirements repositories to manage and track requirements through the life cycle. Case studies and software projects will be required.

S E 412: Formal Methods in Software Engineering

(Cross-listed with COM S, CPR E). (3-0) Cr. 3.

Prereq: COM S 230 or CPR E 310; COM S 311, STAT 330; for graduate credit: graduate standing or permission of instructor

A study of formal techniques for model-based specification and verification of software systems. Topics include logics, formalisms, graph theory, numerical computations, algorithms and tools for automatic analysis of systems. Graduate credit requires in-depth study of concepts.

S E 416: Software Evolution and Maintenance

(Cross-listed with CPR E). (3-0) Cr. 3.

Prereq: COM S 309

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.

S E 417: Software Testing

(Cross-listed with COM S). (3-0) Cr. 3.

Prereq: COM S 309; COM S 230 or CPR E 310; ENGL 250, SP CM 212

Comprehensive study of software testing, principles, methodologies, management strategies and techniques. Test models, test design techniques (black box and white box testing techniques), test adequacy criteria,integration, regression, system testing methods, and software testing tools.

S E 419: Software Tools for Large Scale Data Analysis

(Cross-listed with CPR E). (3-3) Cr. 4.

Prereq: CPR E 308 or COM S 352, COM S 309

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.

S E 421: Software Analysis and Verification for Safety and Security

Cr. 3. F.S.

Prereq: COM S 309; CPR E 310 or Com S 230

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.

S E 490: Independent Study

Cr. arr. Repeatable.

Prereq: Senior classification in software engineering

Investigation of an approved topic.

S E 491: Senior Design Project I and Professionalism

(2-3) Cr. 3.

Prereq: S E 329 and S E 339, CPR E 308 or COM S 352, ENGL 309 or ENGL 314

Preparing for entry to the workplace. Selected professional topics. Use of technical writing skills in developing project plan and design report; project poster. First of two-semester team-oriented, project design and implementation experience.

S E 492: Senior Design Project II

(1-3) Cr. 2.

Prereq: S E 491

Second semester of a team design project experience. Emphasis on the successful implementation and demonstration of the design completed in S E 491 and the evaluation of project results. Technical writing of final project report; oral presentation of project achievements.

S E 494: Software Engineering Portfolio Development

Cr. R. F.S.

Prereq: Credit or enrollment in S E 491

Portfolio assessment for Software Engineers. Guidelines and Advice to improve software engineering portfolios and to better use portfolios as a tool to enhance career opportunities.