

**CAL POLY HUMBOLDT**  
**University Senate**

**Resolution to Recommend A New Bachelors of Science in Software Engineering**

33-21/22-ICC – April 26, 2022 – Curriculum Reading

**RESOLVED:** That the University Senate of Cal Poly Humboldt recommends to the Provost that the new Bachelors of Science in Software Engineering detailed in proposal [21-1488](#) be approved.

**RATIONALE:** The proposed Bachelors of Science in Software Engineering program is set to be housed in the Department of Computer Science. Through this program, students will apply engineering concepts to software development, operation and maintenance of programs. The curriculum of software engineering programs includes computing fundamentals, software design and construction, requirements analysis, security, verification, and validation; software engineering processes and tools appropriate for the development of complex software systems; and discrete mathematics, probability, and statistics, with applications appropriate to software engineering. The ICC and the Department of Computer Science believe this program will prepare students for high demand occupations in software development in fields including business, communications, transportation, medicine, aeronautics, and more as well as careers in cybersecurity. There are six new courses being proposed to support this program. This program is being proposed as part of the transition to a Polytechnic University.

**Bachelors of Science in Software Engineering Program Learning Outcomes**

1. Identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.
2. 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.
3. Communicate effectively with a range of audiences.
4. 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.
5. Function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives.
6. Develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions.

7. Acquire and apply new knowledge as needed, using appropriate learning strategies.

## **Bachelors of Science in Software Engineering Curriculum**

### **Prerequisite (0 – 6 units)**

Students may demonstrate calculus readiness by achieving an appropriate score on a department administered placement test, by successful completion of a course in precalculus, or by completing one of the following prerequisite course pathways, or their equivalent:

MATH 102 (4) Algebra and Elementary Functions, OR

MATH 101 (3) College Algebra AND MATH 101T (3) Trigonometry

### **Lower division (46 units)**

CS 111 (4) Computer Science Foundations I

CS 112 (4) Computer Science Foundations II

CS 211 (4) Data Structures

CS 243 (4) Architecture

CS 251 (4) Requirements Engineering

MATH 107 (3) Introduction to Linear Algebra

MATH 109 (4) Calculus I

MATH 110 (4) Calculus II

MATH 253 (3) Discrete Mathematics

STAT 109 (4) Introductory Biostatistics

Physical Science (4) Lower Division General Education (B1)

Life Science (4) Lower Division General Education (B2)

### **Upper division (37 units)**

CS 312 (4) Algorithms

CS 374 (4) Operating Systems

CS 325 (4) Database Design

CS 328 (4) Web Applications Using Databases

CS 356 (3) Software Design

CS 357 (4) Software Project Management

CS 375 (4) Systems Programming

CS 453 (3) Software Quality Assurance

CS 449 (4) Computer Security

CS 459 (3) Software Engineering Senior Project

## **Electives (6 units)**

At least two advisor-approved special topics courses in Computer Science or a related discipline.

## **Descriptions of New Courses Proposed as Part of the Bachelors of Science in Software Engineering**

CS 251 Requirements Engineering (4) Discuss methods for clearly specifying and managing software requirements. Utilize formal requirement syntax, semantics, and theory. Complete team project(s). Produce technical documents and make formal presentations.

CS 356 Software Design Engineering (3) Learn and practice software development lifecycle: secure design, modeling, architecture, verification and validation. Complete team projects, produce technical documents, and make formal presentations.

CS 357 Software Project Management (4) Learn principles and practices of project management, including resource management, risk management, project manager responsibilities, and team building. Complete team projects, produce technical documents, and make formal presentations.

CS 375 Systems Programming (4) Develop and debug complex programs using debuggers, profilers, source version control systems, and integrated development environments. Analyze execution on modern computers. Measure and optimize performance. Complete team projects, produce technical documents, and make formal presentations.

CS 453 Software Quality Assurance (3) Learn to assess and assure software quality with respect to client needs and software requirements specifications. Hands-on experience through field activities including client consultation. Complete team projects, produce technical documents, and make formal presentations.

CS 459 Software Engineering Senior Project (3) Apply techniques from previous courses while working with actual clients to develop robust, ethical, and effective software. Provide solutions for environmental and sustainability issues as well as cultural and socio-economic problems. Class is enhanced by seminars, tutorials, field activities, and client consultation, and culminates in a significant team project.

