

COURSE DESCRIPTION

Dept., Number	CSci 487	Course Title	Senior Project
Semester hours	3	Course Coordinator	Cynthia B. Zickos, Instructor

Current Catalog Description

Each student conducts an in-depth study of a current problem in computer science or related area. Upon completion, the student presents the results in both oral and written form.

Textbook

None.

References

Course Outcomes

Senior Project represents the capstone experience in the BSCS program (and the BA computer science major) and so is usually taken during a student's last semester on campus. Its outcomes therefore mirror most of the Program's Outcomes. Specifically, students in CSci 487 should, in the course of the semester, demonstrate:

1. An ability to apply knowledge of computing and mathematics that are appropriate to the discipline (Program Outcome a)
2. An ability to analyze a problem and to identify and define the computing requirements appropriate to its solution (Program Outcome b)
3. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet requirements (Program Outcome c)
4. An understanding of professional, ethical, legal, security, and social issues and responsibilities (Program Outcome e)
5. An ability to communicate effectively with a range of audiences (Program Outcome f)
6. An ability to analyze the local and global impact of computing on individuals, organizations, and society (Program Outcome g)
7. An ability to use current techniques, skills, and tools necessary for computing practice (Program Outcome i)
8. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design decisions (Program Outcome j)
9. An ability to apply design and development principles in the construction of software systems of varying complexity (Program Outcome k)

Relationship between Course Outcomes and Program Outcomes

Please see previous item.

Prerequisites by Topic

6 hours of CSci 300+ classes

Major Topics Covered in the Course

As a capstone course, CSci 487 has no required lecture content. Class time is used for discussions of the software development process, for discussion and solution of problems the students are encountering, and for periodic presentations by the students of their project statuses. Near the end of the semester, the students present their projects orally to their classmates, to faculty members, to sponsors, and to other guests.

Assessment Plan for the Course

Students deliver project documents (requirements documents, analysis documents, design documents, finished code, user and systems manuals, software license agreements, and so forth) for grading on a schedule established by the faculty member who teaches CSCI 487 in a given semester. These deliverables are graded for content, for design quality, for grammatical correctness, and so forth.

Throughout the semester, students orally present progress reports throughout the semester, culminating in a public presentation. Using a rubric, faculty members evaluate whether each student completed a project of satisfactory complexity in at least one area of Computer Science and whether the student completed the project satisfactorily. In addition, the faculty members rate the quality of the presentation itself.

How Data in the Course are Used to Assess Program Outcomes (unless adequately covered already in the assessment discussion under Criterion 4)

As explained in Chapters 2, 3, and 4 of the Self-Study, the faculty created standardized tests for ten core courses taken by all Bachelor of Science in Computer Science students. They then drew five representative questions from each test to form another 50-question exit exam that is administered near the end of each offering of CSCI 487. Faculty members consider performance on the exit exam in concert with data collected from the core courses.

Students in CSCI 487 also complete an exit survey that measures their perceptions of their abilities to develop solutions to nontrivial computing problems, to present technical information to a range of audiences, and to work effectively in a team.

Estimate Curriculum Category Content (Semester hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms			Software design		3
Data structures			Concepts of programming languages		