CSCI Course Descriptions

Kevin Buffardi

October 2014

CSCI 430 Software Engineering

3.0 FA WP
Prerequisites: CSCI 311 for CSCI/CINS/APCG majors or EECE 337 for Engineering majors; ENGL 130 or JOUR 130 (or equivalent) all with a grade of C- or higher.
An overview of software engineering principles, practices, and tools. Topics include: agile software engineering methodologies, requirements engineering, test-driven development, software design patterns, MVC architecture, version control, software metrics, and static analysis. Students work in groups to design and implement a semester-long open source software project.
2 hours discussion, 2 hours activity. This is an approved Writing Proficiency course; a grade of C- or better certifies writing proficiency for majors.

CSCI 431 Usability Engineering

3.0 SP
Prerequisites: Junior Standing; CINS 110 with a grade of C- or higher for CSCI/CINS majors, or HTML/CSS/JavaScript experience for other majors.
A study of user experience (UX) with an emphasis on user-centered design and usability evaluation methods. Students practice hands-on techniques including: usability testing, survey design, card sorting, contextual inquiry, wireframing and rapid prototyping. Students work in multi-disciplinary teams on user experience design projects.
2 hours discussion, 2 hours activity.

CSCI 533 Software Testing and Quality Assurance

3.0 SP
Prerequisites: CSCI 430 or CSCI 630 and classified graduate standing.
In-depth study of software verification and validation with a concentration on software testing methods, tools, and metrics. Topics include Test-Driven Development, unit testing, integration testing, acceptance testing, as well as software and testing metrics. Students will work individually as well as in teams to verify and validate large open source projects.
3 hours discussion.

CSCI 630 Software Engineering

3.0 SP
Prerequisites: CSCI 430 and classified graduate standing.
An advanced study of software engineering concepts and practices with an emphasis on software design patterns, static and dynamic analysis, and maintenance of large projects. Teams of students will collaborate on large open source software projects using leading software engineering tools.
3 hours discussion.