

COURSE DESCRIPTION

Department and Course Number: CS 481 **Course Title:** Senior Capstone Design

Total Credits: 4 cr

Course Coordinator: William S. Junk

URL:

<http://www.cs.uidaho.edu/~billjunk/courses/cs481/cs481.html>

Current Catalog Description: Application of formal design techniques to development of a large computer science project performed by students working in teams. Significant lab work reqd. Prereq: CS 383 or 480

Textbook: None

References: Numerous items of reference material are available from the course web site <http://www.cs.uidaho.edu/~billjunk/courses/cs481/cs481.html>. Several types of material are available: project management guidelines, technical documentation guidelines, tutorials, checklists, and some suggestions for functioning as an effective team.

Course Goals: The primary goal of this course is to provide each student with the opportunity to experience and practice the development of a non-trivial, realistic application program from the initial specification to the final testing and deployment in a collaborative, team-based environment.

Projects are sponsored by industry practitioners and university faculty. Each student must learn the skills needed for effective collaboration and in concert must develop the ability to plan, communicate, and coordinate his/her individual contribution to the project so that it supports the overall project plan developed by the team. This course uses a mentoring approach to build on the foundation established by successful completion of CS 383 or CS 480. Emphasis is placed on learning and applying a systematic software development approach based on incremental / evolutionary development. Additional goals are to place students in a situation that will require them to apply and improve their technical writing (IV-16) and presentation (IV-15) skills, to increase their competence in the use of software development tools, and to deliver a thoroughly tested application satisfying the user's requirements, possibly in an environment or on a platform they have not previously experienced. In the context of their specific project, the course addresses social and ethical considerations associated with software product development (IV-17). Some projects may leverage or use code developed by others, requiring appropriate consideration and recognition of intellectual property rights. During the semester each student will have his / her turn to be the project lead, and at another time the project recorder. Each student is also expected to make a significant, reasonable, and fair share contribution to the writing and maintenance of documents, as well as to the development and testing of product code.

Unlike the majority of courses in the CS program, numerical grades are not given on individual work products. Rather, each work product is reviewed for adequacy. Feedback and specific directions are given to the team on what was done well and what needs additional work. Refinement of each work product continues until the work product is acceptable. At the end of the semester, individual grades are given based on each person's level of effort, demonstrated competence, and overall contribution to the success of the team.

Prerequisites by Topic:

- Knowledge of fundamental material covered in the core content areas sufficient to allow achievement of project objectives (IV-6)
- Knowledge of problem analysis and solution design processes and techniques (IV-7) (CS 383 or CS 480)
- Technical writing skills (Engl 317 and CS 383 or CS 480)
- Formal presentations skills (Comm 101 and CS 383 or CS 480)
- Proficient programming skills in a high level language such as C++ or Java (IV-8) (CS 120, 121, 127, 383 or 480)
- Knowledge of basic software development tools found on Unix platforms (CS 240, 383 or 480)
- Ability to understand code written by others (CS 120, 121, 383 or 480)

Major Topics Covered in the Course: At the beginning of the semester, one class session is held to familiarize students with the development process to be used and the expectations for team and individual performance. After this session, teams of 3 to 5 members are formed. Each team has a different project and customer. For the

remainder of the semester each team meets with the instructor once a week for 45 to 60 minutes. Additional meetings, typically 60 minutes in length, are held with the team, instructor, and project sponsor participating. At a minimum these meetings coincide with the project kickoff, each interim product release, and a final project presentation. Between weekly meetings, e-mail is used extensively to facilitate exchange of information. Each teams also meets as necessary and at their convenience to conduct project discussions. During the semester the following topics will be addressed at an appropriate level in the specific context of each project, however, it is difficult to say how many hours are devoted to each topic since it will vary from team to team based on specific needs of their project and their success in meeting expectations. The correspondence of these topics to the CC 201 CS body of knowledge core topics is shown in parentheses.

- Software development using an incremental approach (SE 4)
- Social and ethical issues (SP 4, SP5, SP 6, SP 7)
- Project planning (SE 8)
- Working with a remote user/customer to identify application requirements (SE 4)
- Communicating with the customer (SE 4, SE 8)
- Status reporting (effort, accomplishments, problems, code turmoil, testing, and defects) (SE 8)
- Creating an Operational Specification (SE 5)
- Developing and documenting a software design (SE 1, SE 2)
- User interface and usability considerations (HC 2, HC 3, HC 5)
- Product implementation and evolution (SE 3, SE 7)
- Traceability (SE 6)
- Configuration management and change control (CVS), build tools (SE 3, SE 4)
- Unit and system testing, test automation (SE 4, SE 6)
- Documenting and managing defects (SE 4, SE 6)

Laboratory projects (specify number of weeks on each): The course is a semester-long laboratory activity (approximately 16 weeks). During the semester in which the course display was compiled, three teams and three projects were completed. One project developed a web-based interface for a Central Pivot Irrigation System research project being conducted by a University of Idaho faculty member, one project developed a Black List / White List E-mail and Span Filter System, and another developed an IDL-OSD Visualization Tool.

Estimate Curriculum Category Content:

	Core	Advanced		Core	Advanced
Algorithms			Data Structures		
Software Design		4	Prog. Languages		
Computer Arch					

The Software Design Category focuses on application of the software design and development process as it applies to the application under development. Although the vast majority of activity in this course is contained within the Software Design category, some attention may be paid to other areas as needed to support effective development of the specific application.

Oral and Written Communications: Each team is required to submit at least 4 written reports of typically 5 to 50 pages or more and to make 5 oral presentations of typically 30 to 60 minutes duration. The vast majority of project documentation is posted to a team’s project web site. The contents of the web site are regularly updated to reflect evolution of the project during the semester. Although the documentation for a specific project may have unique requirements, each project will produce an Operational Specification (software requirements), Software Architectural Description, Software Test Plan, and some form of usage documentation (user’s manual, integrated help system, man pages, or release notes). Additional documentation may be produced to meet unique project needs. Each team member is required to make a substantial contributions to the generation and review of technical and project management material for the project.

Three incremental releases, one or more beta releases, and a final product release are expected. With each incremental release and the initial beta release, and final release, the team is required to make a formal presentation to the course instructor and project sponsor reflecting the status of the project. Release presentations are typically 30 – 60 minutes in duration and most include a product demonstration. Each team member is required to assume lead responsibility for preparation and delivery of at least one formal presentation. Appropriate participation in other formal presentations is expected.

Social and Ethical Issues: Students are expected to perform the development of their project in a professional and ethical manner, including open and honest communication with their course instructor and customer. The team must prepare a report documenting their assessment of the social, legal, and ethical issues they face in the assigned project. The total time devoted to social and ethical issues depends on the nature of the project and whether or not specific situations warrant significant discussion. Social, ethical, and professional issues are discussed in team meetings with the course instructor at appropriate points during the project. (5%)

Theoretical Content: None

Problem Analysis: Students are required to develop an operational specification that thoroughly describes the application characteristics (functional, non-functional, and performance requirements) for the semester's project. They must work with the customer to define and analyze the requirements. They must gain a comprehensive understanding of the customer's problem and what each requirement is requesting as well as how separate requirements may be mutually supportive or may interact. (30%)

Solution Design: Students are responsible for designing, implementing, and testing a complete solution meeting the operational requirements of the project. The design is represented in a Design Specification. At the completion of the project the team must deliver a fully and accurately documented product. (65%)