

**CSC400: Applied Software Development Project
Summer 2008**

SYLLABUS

COURSE DESCRIPTION

This course provides an overview of Software Engineering concentrating on the each of the phases of the software development life cycle. Team approach to analyze, design, and document realistic systems of moderate complexity. Project management methods, scheduling and control, and group dynamics in solving systems problems.

PREREQUISITES: CSC152, CSC250, CSC302

CREDITS : 3

Instructor: Dr. Nancy Ashley

E-mail : nashley@linfield.edu (always the best way to reach me)

Phone : Home: (541) 896-3133 Cell: (541) 914-1986

Office Hours : On class weekends only, by appointment.

TEXT:

Required: All texts used in previous courses, and handouts to be given out in class. It is highly recommended that you **bring with you** your texts and reference materials from previous classes, particularly those pertaining to database design, SQL Server, the systems development lifecycle, project management, UML modeling, and Visual Studio.

Resources: Crystal Reports Tutorials and Documentation

[Online Documentation](#)

http://apollo.cs.linfield.edu/course_materials/dce/csc400/Crystal%20Reports/Crystal%20Reports%20Doc/Crystal%20Report/Docs/default.htm

This is an online documentation to be used as reference source.

SOFTWARE: Microsoft Visio, Microsoft Visual Studio.NET, Microsoft SQL Server

OBJECTIVES

This course fulfills suggested curriculum requirements of ACM and the Computer Science Accreditation Board for Computer Science departments. The course will provide the student with an overview of the tools and techniques of software engineering.

This course is designed to prepare the student to be able to join a team involved in software development projects and quickly become productive in that environment.

This course will provide the student with experience in teamwork by incorporating a major project that is to be completed by a group of students. This project will involve significant writing, programming, as the students design, implement, and test their solutions.

Finally, this course serves as a capstone for the BIS curriculum by allowing the student to pull together many of the ideas from other courses as they attempt to find a good solution to their project.

Upon completion of this course the student will:

- have experience in software development activities from specification to testing and maintenance
- have experienced the difficulties of working in teams and investigated techniques for overcoming those difficulties.
- have presented technical material to a group.
- have created precise and informative documents for each stage of software development.
- have applied several approaches to software design such as data flow oriented, data structure oriented, and object oriented techniques.
- understand the importance of quality assurance and reliability of software systems.

Educational Objectives

- Foster group-based software design skills by working in project groups
- Broaden existing software design and programming skills using current methodologies.
- Enhance existing software document skills
- Incorporate design-for-maintainability considerations through appropriate project work

Professional Objectives

- Exposure to design and programming technology that is widely-used in industrial and research environments.
- Enhance planning and time management skills by making students work through a significant project with multiple, staged deliverables.
- To provide a practical capstone experience that will prepare graduates to lead large software projects in industry.

COURSE OVERVIEW

- Specify requirements using Use Cases, Object, Interaction and Dynamic Models

- Build an object oriented solution based on a suitable target system architecture
- Specify, design, build and deliver a two-tier client/server application system.

COURSE PROJECT

During the semester you will work on a group project which will constitute the main activity for this course. More details on course deliverables, and the timing of those deliverables, will be provided in the first class session.

GRADING

Grading for this course will be based mainly on deliverables for the group project, as well as a few individual activities. Because this course is an upper-level course, the standards by which you are assessed and the grading scale used are more rigorous than those in lower division courses. Grading will be based on a curve. More details on grading will be provided in the first class session.

INCOMPLETE

A grade of incomplete (I) is given **only** in emergency situations. The student must request an Incomplete in writing and must obtain my permission. All uncompleted work must be completed within the time limit set by the college. If you simply don't turn in assignments or deliverables for your project, your course grade will be calculated with the missed portion counting for 0 points.

ACADEMIC HONESTY

Cheating and plagiarism will not be tolerated. Any student found engaging in either of these activities at any point in the course will receive a failing grade for the entire course and may be subject to further college sanctions.

RETURN OF COURSE MATERIALS.

Deliverables for the course projects remain the property of the department. If you wish to have a graded copy returned to you, you must submit two copies and provide me with an appropriately-sized self-addressed stamped envelope.

CLASSROOM, GROUP DISCUSSIONS, AND WORKING AT A DISTANCE

Active involvement of each student in class and group discussions is essential for a course of this kind. Students are expected to make pertinent and substantive contributions to every group discussion because team work plays important role in software development projects. We must assume that we are all persons of intelligence and good will who are here to learn from each other in a team environment. Group discussions should not be a forum to impose our ideas on others. For the academic endeavor to succeed, we must treat each other with civility, courtesy and respect. Software development involves team work and all contributions by group members should be discussed and analyzed thoroughly.

It is very important that the teams take the time on the first weekend to get a good feel for the project, and work face-to-face as a team to come to grips with the size of the project, establish communication procedures, and produce a couple of deliverables which are due the first Saturday and Sunday, June 14-15.

After the first weekend students will be doing a lot of development work remotely. It is possible that after the first weekend teams will decide that they do not all need to meet again face-to-face the second weekend of the course. However, it is imperative that all members of each team be willing to spend the time necessary the first weekend to understand the project, and not pressure their fellow team members to leave early. That kind of behavior would cost a person team interaction points. Therefore I suggest that students plan to spend both Saturday and Sunday in McMinnville the first weekend, and be open to what your team decides for the future weekends.

That being said, it is also possible that your team WILL decide that they understand the process and project adequately before 5 pm Sunday. In that case, it is acceptable for students to leave early, IF their entire team agrees on it.

I will be giving a small, rewarding, interesting and painless extra credit opportunity on Sunday morning, as something of a reward for those who choose to stick around to firm up their project.

I do understand that for those of you traveling some distance, this may involve you making a reservation for a motel room Saturday night, and not being sure whether you will need it. I believe you will know early enough on Saturday whether your team plans to work together on Sunday, so that you could cancel your reservation and leave Saturday night if that works for you and your team.

More details and guidance on decisions about working as a team at a distance will be provided at the first meeting, Saturday, June 14th.

DIFFICULTIES

If you find you are having problems with the class – the use of the software package, case tool, attendance, keeping up with the reading, fitting into a group, please let me know. I am always available to help you, but I have to know about the problem while it's going on and before the end of the course. The last few weeks of the course is not the best time to ask for help.

COURSE OUTCOMES

The student should be able to:

- work in a team to deliver a software solution to a business problem
- identify the various phases of a development life cycle – requirements analysis, analysis phase, design phase, coding phase, testing phase
- utilize various techniques in gathering information about the systems functionalities
- capture system requirements using object-oriented concepts, including use cases, classes, objects, attributes, operations, relationships and multiplicity
- create sequence, state, and collaboration diagrams with a visual modeling tool
- design & implement a database to support an application
- apply appropriate user interface strategies to a system
- use the design models to plan and coordinate a user interface
- select proper screen-based controls for input attributes that are to appear on a GUI input screen
- understand the purpose and effectiveness of various testing strategies
- document the software development process