CPS492 - SOFTWARE DEVELOPMENT PROJECT

Professors:	Russell C. Bjork	Spring Semester, 2016-2017
	Russ Tuck	Mondays 4:30-6:00 pm Location TBD
		http://www.cps.gordon.edu/courses/cps491-2
Office:	(Bjork) KOSC 242 x4377	russell.bjork@gordon.edu
	(Tuck) KOSC 243 cell: 408-335-0890	russ.tuck@gordon.edu
Hours:	TBD	

PREREQUISITES: CPS491 and a departmentally-approved Senior Project proposal

COURSE OBJECTIVES:

This course is a continuation of CPS491 for students in the General Concentration, and has similar overall goals. Tied in with these goals, this seminar will serve as a context in which you can complete the Senior Project required for your Computer Science major.

TEXT: Bruegge, Bernd and Allen H. Dutoit. *Object-Oriented Software Engineering: Using UML, Patterns, and Java*. (3rd ed) (Upper Saddle River, NJ: Prentice Hall, 2010)

RECOMMENDED: Brooks, Frederick P. Jr. *The Mythical Man-Month*. (Anniversary ed.) (Reading, MA: Addison-Wesley, 1995).

ON RESERVE: Brooks, Frederick P. Jr. *The Mythical Man-Month*. (Anniversary ed.) (Reading, MA: Addison-Wesley, 1995).

Schach, Stephen R. *Object-Oriented Software Engineering* (New York: McGraw Hill, 2008)

Shneiderman, Ben. *Designing the User Interfac*e. 5th ed (Reading, MA: Addison-Wesley, 2010).

COURSE TECHNIQUE AND PROCEDURES:

The structure of this course is similar to CPS491. Again, the course will have a three-fold thrust, though not the same as that of CPS491:

1. Gaining familiarity with software engineering concepts through reading, oral presentations, and class sessions.

2. Learning how to review others' code and benefit from reviews of yours.

3. Using the implementation, testing, and documentation of your senior projects as case-studies for applying software engineering concepts. Each student or team of students will present working versions of portions of their project and excerpts of the their code to the class several times during the development process.

COURSE REQUIREMENTS AND EVALUATION:

1. Reading and Discussion. For most sessions, you are asked to read material in the main text by Bruegge, the recommended text by Brooks, and/or supplementary materials. In the case of reading in the Bruegge text, you should also look over the "Exercises" at the end of the chapter.
   For each reading assignment, you must prepare a summary of the major points (1-2 pages), with particular emphasis on how the reading applies to the project you are working on, and must give 2 or 3 questions you would like to see discussed in class. These may be used as a springboard for class discussion, and must be turned in at the end of the class for evaluation by the professor. (If the assignment involves more than one chapter, you should summarize each chapter individually, but prepare one set of questions for the entire assignment. Summaries for chapters from Brooks can be quite brief. The length specification is overall length)

   Once during the semester, you and one other student will be responsible for leading the class session on one of the topics. There will be an opportunity to sign up for specific topics at the first class meeting.

   Your journals and class discussion leading, together, will account for 20% of the grade.
2. There will be four demonstrations of your project in class. The first two are prototypes, the third is the complete product ready for full customer use, and the fourth is the improved product resulting from observing the customer's use and listening to their comments.

   First demo: Since the initial prototype last semester had a critical part of the core system working, the first prototype demo this semester should have a clear majority of the system working. The most common use cases should be working. You should show this to your customer and get feedback.

   Second demo: The second prototype demo should be essentially complete, with all use cases implemented. There may still be some rough edges to fix and corner cases to implement. You should let your customer try using it, observe them, and get their feedback.

   Third demo: Demonstrate that your system is complete and ready for full use by your customer. Give it to them, watch them use it, and listen to what they say (and don't say).

   Final demo: Demonstrate your final system, focusing on how your system has improved as a result of what you learned from your customer's use.

3. All code in your project should be reviewed by someone other than the author, using GitHub pull requests.

   Each student should select one good example of a significant code review which led to code changes, and share it with the class. The review should cover a non-trivial (more than a few lines) but readable (not much over a page) chunk of new code. Email a link to the pull request (with the review conversation and code changes) to the class no later than midnight Thursday before the class where we will discuss them. Include in the email a brief explanation of how the code under review fits into the overall project, and what the goal of the pull request was. In class, very briefly summarize the change and the review, then answer questions. Your team will be graded on the quality of the review. You will be graded individually on the quality of your email and oral summary, and your comments on other people's reviews when we discuss them.

4. Your team will be assigned to work with another team on the testing of their project - i.e. you will be do system testing of that team's project, and some other team will do system testing of yours. (This mirrors the practice found in many companies of having a quality assurance team assigned to a project that is separate and distinct from the development team, though the two teams work together on the same project.)

   Based on a study of the documentation for the project you have been assigned to, you will prepare and execute a functional test plan, turning in both a copy of the plan and a test report, and you will make an oral presentation to the class on your work. In addition, you will give a copy of your test report to the student/team whose project you tested, so that they can utilize your findings to improve the final product.

These documents and presentations will count in the course grade as follows:

Document	Document Weight	Oral Presentation Weight
First (prototype) demo (80% functionality, 20% presentation)		5%
Unit tests		5%
Second (prototype) demo (80% functionality, 20% presentation)		10%
Third (Complete System) demo (80% functionality, 20% presentation)		10%
Documentation and Tests (Project Notebook on 3/20)	10%	
Code Review (document is code review; oral is email, summary, and comments on other reviews)	6%	4%
Test Plan/Report for another team's project	10%	5%
Completed Project	25% *	10%

* Based on conformity to specifications; operation, user-friendliness, etc; and internal documentation and code quality.

In preparing your presentations and demonstrations, please note the following:

• Each prototype demonstration and test report should be prepared for fellow technical personnel (the class), but your final presentation should be prepared for an audience including people from outside the class. If at all possible, your project "client" (as well as your parents and friends) should be invited to the final presentation; and other CS students will be invited to this presentation as well.

• If your project is being done by a team, each member must participate in each presentation in a roughly equal fashion.

• Each prototype demonstration and the Test Plan/Report presentations should be planned for about 10 minutes each. In each case, this may mean that you will have to limit the scope of your presentation. It is better to cover a few aspects of your project well than to attempt to cover all aspects poorly. Your written code and documentation will show the professor how much you have actually accomplished, and will be the prime basis for evaluating the scope of your work. The purpose of the oral presentations is to give you practice with some of the kinds of presentation you may be called on to make in industry. **Oral presentations improve with practice. Please practice yours at least twice before coming to class.**

NOTE WELL: ALL REQUIRED DOCUMENTS MUST BE TURNED IN COMPLETE AND (IN THE CASE OF FINAL VERSION OF THE PROJECT) WITH THE NECESSARY SIGNATURES IN ORDER FOR YOUR WORK TO BE CONSIDERED COMPLETE. YOU CANNOT RECEIVE A PASSING GRADE IN THE COURSE IF ANY REQUIRED ITEM IS NOT SATISFACTORILY COMPLETED.

EXTENSIONS AND INCOMPLETES:

By college policy, spring semester incompletes are **NOT** possible for seniors graduating in May. Thus, if it appears by mid semester that an incomplete will be necessary, then your graduation will have to be postponed until September. The department may postpone your graduation until September if you miss the milestone scheduled for the Monday after spring break. This is for your protection, since a postponed graduation is better than the risk of having to take an F for the course if your project is not completed by graduation. (September graduates may petition to participate in the graduation exercises with their class in May; but May graduates who fail must retake the course and cannot graduate until the next year.) If an incomplete becomes necessary, it must be made up by July 1 (for September graduation) or October 1. Otherwise, it will automatically become an F. Please note that, if you get behind early in the semester it will be very difficult for you to finish your senior project in time to graduate in May. You MUST keep up!

ATTENDANCE POLICY:

Attendance at all sessions is mandatory. Each unexcused absence will result in a reduction of 1/2 of a letter grade in the term grade.

ACCOMMODATION FOR STUDENTS WITH DISABILITIES:

Gordon College is committed to assisting students with documented disabilities (see Academic Catalog Appendix C, for documentation guidelines). A student with a disability who may need academic accommodations should follow this procedure:

1. Meet with a staff person from the Academic Support Center (Jenks 412 X4746) to:
   a. make sure documentation of your disability is on file in the ASC,
   b. discuss the accommodations for which you are eligible,
   c. discuss the procedures for obtaining the accommodations, and
   d. obtain a **Faculty Notification Form**.

2. Deliver a Faculty Notification Form to each course professor *within the first full week of the semester;* at that time make an appointment to discuss your needs with each professor.

Failure to register in time with your professor and the ASC may compromise our ability to provide the accommodations. Questions or disputes about accommodations should be immediately referred to the Academic Support Center. See Grievance Procedures available from the ASC.

TENTATIVE COURSE SCHEDULE - SUBJECT TO CHANGE:

	UNIT IV: IMPLEMENTATION	
M 1/23	Unit and Integration Testing (Select student-led session topics for the semester)	Bruegge ch. 11 through p. 471
M 1/30	Further Demonstration of Working Prototypes	**5-10 UNIT TESTS PER STUDENT (EACH TEAM MEMBER DOES OWN) DUE**
M 2/6	User Interface Design (Student-led session)	Shneiderman (on reserve) ch. 1, 4
M 2/13	Code Reviews	**EACH TEAM MEMBER IDENTIFY A GOOD REVIEW OF HIS/HER OWN CODE THAT RESULTED IN CHANGES. EMAIL A LINK TO THE GITHUB CODE REVIEW BY THURSDAY MIDNIGHT BEFORE CLASS. READ ALL REVIEWS AND BE PREPARED TO DISCUSS THEM IN CLASS.**
M 2/20	Further Demonstration of Working Prototypes	
M 2/27	System Testing; Documenting Testing. (Student-led session)	Bruegge ch. 11 pp 471-end
M 3/6	Refactoring; Appropriate documentation (Student-led session)	TBD
M 3/13	**SPRING BREAK**	
M 3/20	Demonstration of Complete Product; Test Planning Work Session	**PROJECT NOTEBOOK WITH PRELIMINARY UNIT AND INTEGRATION TESTED, DOCUMENTED, WORKING VERSION OF PROJECT DUE, TO BE SUBMITTED TO THE USER, STUDENT TESTER, AND DEPARTMENT FOR EVALUATION.**

		(Department gets full notebook; student testers get a copy of the the Requirements document and User manual, plus access to the software; user gets access to the software)
	UNIT V: PROJECT MANAGEMENT	
M 3/27	Software Evolution/Maintenance (Student-led session)	Schach (on reserve) ch. 14
M 4/3	Project Management; Starting Over. (Student-led session)	Bruegge ch. 14 Brooks ch. 11 *
M 4/10	Test Plan and Test Report Presentations (Oral Presentations)	TEST PLAN, REPORT DUE; (USER AND DEPARTMENT COMMENTS ALSO RETURNED TO TEAM)
M 4/17	(*Easter Break - no class)*	
M 4/24	Software Lifecycle and Software Engineering Methodologies (Student-led session)	Bruegge ch. 15, 16
M 5/1	Practice Senior Project Demonstrations (Oral Presentations)	**COMPLETE NEAR-FINAL VERSION OF EVERYTHING DUE!**
M 5/8 4:30-7	Senior Project Demonstrations (Oral Presentations)	**FINAL VERSION OF EVERYTHING DUE!**
M 5/15 6-8pm	Dinner party and “post-mortem” discussion during final exam period. **CONGRATULATIONS - YOU MADE IT!**	