SYLLABUS

I. Title: CSC493 Software Project

II. Time and Place: Spring Semester 2015
7:30 – 8:20 a.m. MWF, Virus lab

III. Credit: Three units

IV. Instructor: Dr. McKinstry, Professor of Computer Science

V. Office Hours: Rohr Science 216, (619) 849-2269; email: jeffmckinstry@pointloma.edu

Monday:	8:30-9:20 a.m., 11:00-11:50, and 2:45-3:50 p.m.
Wednesday:	8:30-9:20 a.m., 11:00-11:50, and 2:45-3:50 p.m.
Friday:	8:30-9:20 a.m., 11:00-11:50, and 2:45-3:50 p.m.

VI. Text: Recommended: Astels D, Miller G, Novak M (2002). A Practical Guide to eXtreme Programming. Upper Saddle River, NJ: Prentice Hall PTR.

VII. Objectives of the course: "… the CC2001 Task Force recommends that curricula include a capstone project that allows students to bring together all the skills and concepts that they have previously learned during their undergraduate courses. Such a course might include a small amount of additional material, but the major focus must be on the project." – *Computing Curricula 2001, Computer Science* produced by The Joint Task Force on Computing Curricula, IEEE Computer Society, and the Association for Computing Machinery.

VIII. Learning outcomes:
Students will be able to write correct and robust software.
Students will be able to apply their technical knowledge to solve problems.
Students will communicate effectively orally and in writing.
Students will understand the interaction between hardware and software.

IX. Course Organization: The Course Schedule provides an outline with dates for some of the important activities of the course. Time expectations

1. Working on software project in the virus lab for 50 minutes, 3 times per week during class.
2. Weekly project status meeting with the professor during class on Mondays to discuss project status.
3. An additional 5 hours per week outside of lab time working on the project. The assumption is 2 hours outside of class for every hour in.
4. Record all hours on a time card and turn it in at the beginning of the Monday status meeting. Timecard has the following format:

Date	Start time	End Time	Total time	Description
Jan 13	7:30 a.m.	8:20 a.m.	50 min.	Weighed project options
...				
Total			**8 hours**	

X. Student Evaluation:

Weekly status meeting participation/attendance/timecard	15%
Project Requirements (1 version)	10%
Iteration 1 demo and acceptance tests	15%
Iteration 2 demo and acceptance tests	15%
Iteration 3 demo and acceptance tests	15%
Iteration 4 demo and acceptance tests	15%
Customer's grade based on final product & code quality	15%

You will be required to demonstrate your programs to the professor and to turn in an electronic copy of your work.

Grades will be determined as follows:

93-100%	A
90-92%	A-
87-89%	B+
83-86%	B
80-82%	B-
77-79%	C+
73-76%	C
70-72%	C-
67-69%	D+
63-66%	D
60-62%	D-
0-59%	F

XI. Course Schedule (subject to change).

Week 1, Jan. 12: Pick projects. Project preference ranking due Wednesday. I will assign projects by Friday.
Week 2, Jan. 19: Team role selection Monday + user stories due Friday
Week 3, Jan. 26: project status meetings Monday + story point estimates due Friday
Week 4, Feb. 2: project status meetings Monday + weekly build demo Friday
Week 5, Feb. 9: project status meetings Monday + weekly build demo Friday
Week 6, Feb 16: project status meetings Monday + **iteration 1 acceptance tests Friday.**
Week 7, Feb 23: project status meetings Monday + weekly build demo Friday
Week 8, Mar. 2: project status meetings Monday + weekly build demo Friday
Week 9, Mar. 9: **Spring Break**
Week 10, Mar. 16: project status meetings Monday + **iteration 2 acceptance tests Friday.**
Week 11, Mar. 23: project status meetings Monday + weekly build demo Friday
Week 12, Mar. 30: project status meetings Monday **+ Easter Recess Friday**
Week 13, April 6: **Easter Recess Monday** + project status meetings Wednesday + **iteration 3 acceptance tests Friday**
Week 14, April 13: project status meetings + weekly build demo Friday
Week 15, April 20: project status meetings + weekly build demo Friday
Week 16, April 27: project status meetings + **Final product acceptance tests Friday**
Week 17, Finals week. Extra time for iteration 4 demos Monday, May 4th, from 7:30 to 10:00 a.m. (if necessary).

XII. Projects

The project is a group (or possibly an individual) project. All projects must have a customer who cares about the project and wants to use the software after the course is over.

Project #1: Neuroscience research (Applies only to juniors who plan to consider doing summer research with Dr. McKinstry in summer 2015). If interested, discus with Dr. McKinstry after class. One current project involves developing and testing software that can recognize arbitrary motion paths of 3D objects. This may involve learning to program in MATLAB. Obviously the first month will involve background reading.

Project #2: Robot simulation software. There is a software simulator, called Player Stage, that allows one to simulate various commercial robots in a 3D virtual world. The project would involve installing this system on a Mac or PC, and writing code in C++ to control a virtual robot to attempt to solve one of the tasks from the DARPA ARM project. You can search the web for more info on this. This would be a lot of fun, but from past experience, it is VERY challenging.

Project #3: Artificial natural selection of a retina neural network. The idea is to write software that simulates natural selection in an artificial environment in order to "evolve" a neural network that resembles animal retinas. The computation of the retina network is well-known, so we have a good idea of what to look for. To my knowledge, no one has done this, and I believe it would be a significant achievement. Furthermore it could be done by a team of undergraduates.

Project #4: Open source project: Join an Open Source software project team of your choosing and make a contribution. You must suggest which project you will join and describe it if you choose this project.

Project #5: Find your own customer: You may find a customer of your own that has a software need and write software for them. This must NOT be a job you are already paid to do, and may not be a project you are doing as part of another course. The customer must contact me, and write a brief project description by the end of the first week.

Project #6: Simulated robot hand. Simulated robotic arm and hand in virtual world. This project involves teaching a custom robot avatar in the Webots software environment to pick up small objects using only the sense of touch. Real robotic hands do not currently have skin which can sense contact with objects. DARPA estimates that robot hands with skin are at least 4 or 5 years away, therefore, simulations are the only option at this point. Students should have some experience with neural networks or be signed up for special topics this semester which covers AI and neural networks.

Project #7: Continuation of software engineering projects from last semester. Since you developed some expertise, some of the team may want to continue to perfect the project. The goal would be to deliver a finished product that is robust and make it available to the world.

Project #8: Apply deep learning toolkit (Caffe or equivalent) to solve one of several machine learning challenges. Machine learning challenges include labelling humans in motion in video clips. Website with more info on the competition will be provided. You should be signed up for special topics (AI) this semester for this project.

Project #9: Robot visual localization. Students will use a Roomba robot with a laptop aboard to learn where it is in a room from prior exploration. Students will have to learn Matlab, and should be signed up for special topics this semester.