

**CSCE 487**  
**Computer Science Professional Development**  
**Class Syllabus**  
**Spring 2011**

<b>Instructor</b>
-------------------

Matt Dwyer, Avery Hall 365  
Office Hours: by appointment  
Email: dwyer at cse.unl.edu

<b>Course Webpage</b>
-----------------------

<http://cse.unl.edu/~dwyer/487>

<b>Requirements</b>
---------------------

**PreReq:** CSCE 361, must be taken exactly one semester after CSCE 486

<b>Course Objectives</b>
--------------------------

- Practice with a significant design project in a realistic design environment, to gain mastery of: open-ended problem solving, design skills, written and oral presentation skills, and team dynamics,
- Practice in the application of standards, regulations and professional ethics, as allowed by the context of the project.

<b>Relationship to ACE</b>
----------------------------

**This course will satisfy ACE Learning Outcome #10:** Generate a creative or scholarly product that requires broad knowledge, appropriate technical proficiency, information collection, synthesis, interpretation, presentation, and reflection.

**We will provide the following opportunities for achieving the above ACE Learning Outcome.** This course is the capstone course for our Computer Science curriculum, taken by our Computer Science seniors nearing graduation. The course provides intensive opportunities for hands-on practice, close to a real-world environment in terms of teamwork, problem identification, and research-and-development processes, requiring students to gain mastery in open-ended problem solving and design skills, written and oral presentation skills, and team dynamics.

This course satisfies the ACE Learning Outcome #10 through a very specific requirement: each team of students must conceive, design and build an innovative hardware-software device of sufficient complexity so as to require effective teamwork. To achieve this goal, they must follow the critical steps involved in computer- system prototyping: (1) integrate knowledge ranging from theoretical to practical, mathematical to computational to logical, acquired during students' study here, to address a wide range of design issues and challenges from software to hardware (**broad knowledge**), (2) design, plan, and execute the design with technical proficiency in searching and identifying suitable computer and hardware parts, system assembly, fabrication,

testing, evaluation, and refinement (**appropriate technical proficiency**), (3) perform technical and literature reviews of existing technologies for feasibility and comparative analyses of the design choices (**information collection**), (4) synthesize technical data and specifications of computer and hardware parts (such as power consumption, pricing, and performance charts of a radio-frequency (RF) sensor) such that the final prototype meet design constraints, requiring discussions, insights, empirical tests, and trade-off analysis (**synthesis**), (5) interpret and present results in terms of system performance, requirements, and costs, involving extensive component testing of individual modules as well as the final product testing, often involving in-situ, real-time situations (**interpretation and presentation**), and (6) conduct system evaluation and teamwork in terms of lessons learned throughout the design and development process, highlighting paths that led to dead-ends, breakthroughs, and “engineering tweaking” that get the product to work, in both oral presentation and written report forms (**reflection**).

For students’ designs or products to perform well, they must be able to integrate and apply a broad range of computer engineering knowledge as well as problem solving, communication, and teamwork skills. This is why this course is **critical** in our curriculum as the ultimate assessment of how well matriculating students from our program have learned and are capable of applying what they have learned. Furthermore, since all problems to be solved are open-ended, students are forced to **explore and evaluate** possible options, **make** design-critical **decisions**, be **creative**, and be **resourceful**. The best products are often the combination of hard work and imagination, supported by good teamwork and a solid engineering process.

We will **assess your achievement of the outcome** in the following manner. The quality of the final product primarily paints the picture of how well the Learning Outcome has been achieved, as well as the process during which the final product is derived from a problem to a solution, and evolves from a conceptual design to a concrete computer-hardware system that is rigorously tested either individually or in a real-time contest. First, there is at least one fully-functional, end-to-end product demonstration. This will evaluate students’ product **creativity, quality, technical correctness**, the application and integration of computer engineering concepts and paradigms (**broad knowledge**) in order to assess their mastery of computer engineering concepts and paradigms, proficiency in problem solving and product development (**appropriate technical proficiency**), and familiarity with the engineering process such as exploring design choices, decision making after comparative studies, and performing trade-offs when planning for product design and development (**information collection and synthesis**). There are at least two written and oral reports to be written and presented as a team. These will assess the students’ project progress on different stages: from development to testing, from prototyping to final product demonstration. Further, these will assess students’ **information interpretation and presentation** skills. Product evaluation and refinement typically involve extensive experiments and subsequent result analyses. Students’ documentation, discussions, and presentation of results are graded via these written and oral reports. Further, both reports require students to **reflect** on lessons learned, unfinished work, potential extensions, and societal impact of their design and product.

**This course will also reinforce the following skills.** Since this course is project-based and highly open-ended, encountering unexpected problems is the norm and every team must be able to resolve them through **teamwork**. These problems range from: technical issues requiring careful analysis and research for resolution, cost issues because every team must complete the project under their allocated budget, and time issues that arise because of imperfect coordination

of work within a group or due to external factors, such as backordering of a popular part. Dealing with these problems requires students to perform **critical thinking** such as prioritization, resource management, product redesign, and other engineering strategies. The skills are reinforced through reports and presentations as the instructor and other students give feedback on their concepts, designs, solutions, and results. Given the nature of the problem to be solved, students are required to collaborate significantly, utilize their strengths well, assign roles, and manage their own resources well. Teamwork skills are thus reinforced throughout the semester. Students are required to work together in all aspects of the project: from conceptualization to design, from implementation to evaluation, from writing reports to product demonstration. The instructors use standard tools, such as peer and self evaluations both to emphasize **teamwork** and to catch and resolve teamwork issues before it is too late.

### Topics Covered

Precise topics vary from project to project, but include practice in the following:

- Design & implementation of a combined software project too complex for one person,
- At least two written and oral reports to be written and presented as a team,
- Teamwork, collaboration, and professional conduct

### Useful Resources

Many reference materials on oral and written communication exist. (links available at the course webpage):

- Resources for technical writing, Dr. Steve Goddard
- How to write a research paper, Dr. Jean-Yves Le Boudec
- Common bugs in writing, Dr. Henning Schulzrinne
- Style in business writing, Dave Dusseau
- A Handbook for Scholars, Mary-Claire van Leunen
- Little, Brown Essential Handbook for Writers, Jane E. Aaron
- Purdue Writing Lab
- Computers and Society, COS 490, University of Maine

### Instructor's Role

- The instructor will play the part of your "Project Manager". As such, he is not directly involved in the design or implementation of the project. Rather, his role is managerial and advisory.
- You may assume that your project manager came up through the engineering ranks and thus is (or was) a competent computer engineer. However, in the last few years, his time has been consumed more by management duties than by technical duties, so he may be a bit "rusty".
- Furthermore, your project manager has several other projects also under his supervision, so he cannot give complete attention to yours.

### Grading Policies

#### General:

- There will be no formal homework assignments.
- There will be three project reports and presentations.

- Each report/presentation must be a “stand-alone” report/presentation. That is, when you write your report/presentation, you should assume that we know nothing about your project.
- Several times during the semester you will be asked to submit your self and peer evaluations as team members and your design note book (see more details below). These components will be graded individually. Also, there will be an individual component to team oral presentations.

### **First Progress Report (20%):**

- Scheduled approximately during the sixth week of class.
- Each team will submit a report and give a presentation to demonstrate progress on the project and likelihood of its successful completion before the end of the semester.
- The grade for each team will be based on
  - 50% on the technical merit of the design (common grade for the team)
  - 25% on the quality and clarity of the report (common grade for the team)
  - 25% on the quality and clarity of the presentation (individual grades within a team may vary)

### **Second Progress Report (25%):**

- Scheduled approximately during the 11th week of class.
- Each team will submit a report describing your first complete design.
- The report should describe the design concept, analysis of design choices and architecture, implementation, testing, any difficulties encountered, and any shortcomings of the design that would be removed in the final report.
- Each team will also give a presentation, along with a demonstration of their project.
- The grade for each team will be based on
  - 25% on the technical merit of the design (common grade for the team)
  - 25% on the quality and clarity of the report (common grade for the team)
  - 25% on the success of demonstration (common grade for the team)
  - 25% on the quality and clarity of the presentation (individual grades within a team may vary)

### **Final Project Report (30%):**

- In the finals week of the semester, tentatively, at the scheduled time.
- In addition to the requirements for the second progress report, the final report should also analyze the cost of your project (e.g. development cost, equipment cost, and so on)

### **Self and Peer Evaluations (10%):**

From time to time you will be asked to evaluate yourself and other members of your group as part of a team by filling out a form. This feedback will help me judge individual contributions within a team. At the same time, the feedback may help me resolve potential problems in the functioning of a team before they become serious.

### **Weekly Meetings: (10%):**

At every meeting, the team should submit a 1-2 page short report, describing their project progress so far, the goals for the last week, whether you have achieved these goals, and the goals for the next week. Lack of active participation in the weekly meeting, including attendance, may result in a lower grade for individual team members.

### **Design Notebook (5%):**

Each team member should maintain a separate design notebook, like a log book or journal, to record your individual contribution to the project. Your design notebook is like a journal in which you will record your progress through the course. It will be the resource you will refer to for writing your periodic progress reports. The following are specific guidelines that you should keep in mind when making entries into your design notebook:

1. One substantive, dated entry must be made per week describing the progress made during that week. Additional entries may be made to summarize each group meeting and to document work during more intensive periods of activity.
2. The notebook must be neat but should not consume an enormous amount of your time, since it is not for publication. On average, one page per week should be enough. Write and sketch legibly using a pencil (for ease of corrections); don't waste time trying for a publication-quality output. Cut-and-paste whenever it saves you time. You may submit your notebook in an electronic format (e.g. pdf), but be wary of spending too much time doing so. This is especially true for sketches of schematics.
3. Design ideas should be recorded in the notebook *as they occur, not done hastily the day before the due date*.
4. Detailed written descriptions must be made of all designs, with labeled sketches and photos as necessary.
5. Results of testing and subsequent revisions of designs must be included.
6. A written overview of all algorithms (e.g. written summaries, pseudo-code, or flowcharts) and complete, fully-commented listings for all code must be included.
7. Any sensor calibration data that are taken must be recorded.
8. Bring your design notebook to all weekly meetings and submit it at the time of each progress report.

### **Grading**

Final grades will be assigned approximately based on the following scale.

A+	above 97%
A	93% - 96%
A-	90% - 92%
B+	87% - 89%
B	83% - 86%

B	80% - 82%
C+	77% - 79%
C	73% - 76%
C-	70% - 72%
D+	67% - 69%
D	63% - 66%
D-	60% - 62%
F	below 60%

### **Academic Integrity**

Unless specifically prohibited by the instructor, it is acceptable to discuss the meaning of assignments. Discussing general approaches and strategies for solutions may be permissible, but unless specifically allowed, such communications should not include written material or code and should not transmit substantive or specific elements of a solution. In particular, shared flowcharts, pseudocode, code, or documentation are not allowed unless specifically permitted by the instructor. Any cooperation beyond discussing general approaches and general strategies, including shared pseudocode or flowcharts, shared code, or shared documentation, is not allowed unless specifically permitted by the instructor. For more details, please see the CSE Departmental [Academic Integrity Policy](#). Note that copied and lightly edited homework solutions from the Internet or any other source is strictly prohibited.