Projects will be team-based. Each team will develop a project, using the C programming language, that incorporates techniques learned in the class. Projects will be graded on how well the project uses the techniques and how many of the techniques are incorporated in the project. Incomplete projects will not be graded and receive a score of 0.

Each project team will demonstrate their project in a 10 minute demo to the entire class at the end of the term. Each project team will also produce a project report.

Teams are free to propose any project they wish. Projects are expected to evolve to include new techniques as they are studied in the class.

1 Project Teams

Project teams will be assigned by the instructor by the second week of class. Team size are envisioned to be 3-person, but this could change, depending on circumstances.

*Be aware that students may drop during the term.* This could have unfortunate consequences for the team project. If you feel that a member of your team is not performing well or is in danger of dropping the course, inform the instructor at once. **Students who are not sure that they will finish the course should contact the instructor as soon as possible.**

2 The Project

Each project is required to use the techniques learned in the class, to the degree possible. This means that projects will evolve as the class progresses.

2.1 Project Evolution

How you approach the project will have an impact as to how easily new features can be incorporated. How each team chooses to approach this is an internal team matter and not subject to grading. However, doing this wrong could lead to heavy rewrites as new features are incorporated or existing features modified.

One possible approach to development that helps to enable the addition of new features is component based software engineering (CBSE). While described in terms of OO development, the concepts easily map to non-OO languages such as C.

Another, equally good approach is design by interface. This can be seen as a specialization of CBSE.

You are not required to use any approach described here. Use a technique that you already know and with which you are comfortable. No design documents will be turned in for the class; any such documents are only for internal use by team.

2.2 Choosing a Project

The best advice here is to select a project that can be accomplished within 8 weeks. Scan the techniques described in the text and consider what types of features a project could have to utilize those techniques. You are not required to use them all, but part of the grading will be based on which techniques you implement.
Analysis paralysis is the name given to the situation where a team becomes so immersed in designing a feature that it fails to produce the feature. Be very careful not to fall into this trap.

As a project example, perhaps your team wants to develop a multi-threaded shell that includes the ability to chat with other users of the shell and a file transfer feature that uses asynchronous I/O. This means that the shell would need to know how to “talk” to other instantiations of the shell (hint: signals, pipes, sockets). These other instantiations could be on the same machine or, more ambitiously, another computer system. Note that peer-to-peer or client-server models would both work here.

2.3 Project Proposal

Each team is responsible for proposing a project by the last class meeting in the second week of the class. The project proposal may be accepted, rejected, or conditionally accepted with required modifications. If the project is not accepted, the team will have 2 additional days to submit an acceptable proposal. Failure to submit an acceptable proposal will result in the team being dropped from the course.

The proposal does not have to be detailed but rather a broad outline that indicates major features.

2.4 Project Report

Teams will produce a project report. The report will, hopefully, be short; less than 10 pages would be ideal. The report structure is:

1. Introduction. An overview of the project and why it is a good fit for this course.

2. Project responsibilities. This section will outline which parts of the project each team member was responsible for implementing. Be sure that your project is structured so that this section is meaningful.

3. Major features. This section will describe how techniques from the course were used in the project. Do not include source code.

4. Evaluation. This section is a self-assessment by the team. The purpose is to show how well the team utilized the techniques of the class. This section should clearly show for which parts of the project each team member was responsible.

5. Summary. This section provides feedback on the usefulness of the project in terms of learning the course material and also feedback on what the team would have done differently now that the techniques are known.

6. Feedback to the instructor. Feedback that you would like the instructor to know. What worked in the class, what did not work, and what needs improvement. This section will not be graded but will help with future offerings of the course. This section is optional for undergraduate students and required for graduate students.

2.5 Turning in the Project

Project source and report will be turned in on the last day of the term using D2L.