Due Dec. 9, 11.59pm

The term project will attempt to bring together object oriented concepts learnt throughout the semester and will be a group project. Some rough guidelines (these will be updated) include the following:

- Some potential projects will be suggested; you are welcome to use them.
- You can also propose a project, so long it is approved by the instructor.
- The project will be done in teams of 2-3 students; expectations will be higher for the larger groups.
- Projects will be reviewed each week, during class, to review the progress.
- Design documents/Software Architecture for the project will be expected prior to Thanksgiving (Nov 20).

What should the term project comprise of?

- Projects can be from any application domain or discipline.
- Typical projects should lend itself to a reasonable object hierarchy, that can advantage of concepts of object inheritance, polymorphism, virtual functions, overloading (not all have to be satisfied).
- You must be able to test your implementation, preferably via a graphical interface, or an application that uses the object hierarchy.
- Final project will be required to be fully documented, a software architecture of the implementation and a report that details what worked, what didnt work, results, etc.

Project Timeline

Date	Action
Nov. 1	Project Start, Team formation
Nov. 8	Project proposals due
Nov. 15	Initial Review/Design
Nov. 20	Finalized Design/Initial Review
Nov. 29	Project Review
Dec. 4	Final Review
Dec. 6	End of classes/feedback/Review form
Dec. 11	Final Exam Period - Presentations/Demos of Projects

Project Report

Each team must submit a project report. It should contain a detailed description of your project and will contain the following sections:

- **Title, project team members.**
- **Introduction.** Describe what your project is about, what is the motivation or significance. You may also include any related work and possibly references to it (references, if any should be at the end of the report with full citation)
- **Design.** Describe your overall project design, object hierarchy and relationships between different components. Use figures to clearly explain how the whole project is tied together, so that a reader has a clear picture of the whole software design. Explain any choices made in the design and if possible, compare to any alternate design.

- **User Interface/Features.** Describe the user interface of your application (regardless of whether you had a graphical interface or not). Also detail the features of your application.
- **Testing.** In this section, describe how you tested your implementation, what data sets were used, efficiency concerns. In particular, your testing should be robust enough to work with large datasets (if applicable) and possible use of data structures to mitigate this. Use tables or charts to illustrate your testing results.
- **Issues/Status.** Explain the current status of the application and any issues faced, features that are not working or unimplemented, etc.
- **Conclusions**

Evaluation Rubric:

1. **Object Oriented Programming Concepts.** This would include basic concepts of objects, inheritance, inheritance, polymorphism.
2. **Complexity of the Implementation.** This could be measured by object hierarchy, object composition, functionality.
3. **Project Scope.** This will typically be measured as a function of the project time (3.5-4weeks), significance of problem, what was accompished, and work put in by each member of the team.
4. **Completion.** This would focus on what was accomplished vs. project goals and design.
5. **Testing.** How well and thoroughly the project was tested, test cases, the datasets used or generated, case studies used in testing.
6. **Peer Evaluation.** Each member will complete a peer evaluation of the project that will be used to weight the project grade.

Evaluation:

- Each team will be set up as separate group in Canvas.
- Each team will do a short power point style presentation and demo of their project during the final exam period.
- **To Turn in to Canvas:** All source code files, design documents, documentation, Peer evaluation forms.