# 24-780 Engineering Computation Fall 12

## Project

| | | |
|---|---|---|
| **Due:** | Team preference | Due 10/29 (Mon) 23:59 by E-Mail |
| | Conceptual design: | Due 11/05 (Mon) 23:59 @ Course Blackboard |
| | Detailed design: | Due 11/12 (Mon) 23:59 @ Course Blackboard |
| | Component code: | Due 11/19 (Mon) 23:59 @ Course Blackboard |
| | Alpha-version code: | Due 11/26 (Mon) 23:59 @ Course Blackboard |
| | Final-version code: | Due 12/03 (Mon) 23:59 @ Course Blackboard |

**Weight:** **22**% = 1% (Conceptual design) + 1% (Detailed design)
+ 2% (Component code) + 3% (Alpha-version code)
+ 15% (Final-version code & presentation)

You will be designing and developing graphical and interactive software package using C++ for one of three applications:

Education (e.g. digital flashcard)
Entertainment (e.g. shooting game)
Engineering (e.g. robot path planner)

One team can have five to seven students. If everyone in your team agrees on forming a bigger team, you can have more than seven students, but note that everyone in your team must write some code, and the bigger the team, the more difficult to integrate individual codes.

An auditing student is NOT obligated to join a team. However, if a team accepts an auditing student, an auditing student can work on the project. One team can have up to one auditing student, and the work load of the auditing student must be substantially lower than a student registered for a regular grading option.

Each project member, except an auditing student, must be in charge of at least one functional component of the program. A functional component can be either one function or one class. If there is some uneven workload for coding, it is ok as long as each member contributes – a member who takes a light coding can take a heavier load in report writing or presentation.

The system should have two features: a graphical display (2D, 3D, or mixed) and user interactions. The amount of effort you will spend for the project should be three or four times of a regular problem set. You can take some ideas from an existing software program. However, there must be some original idea in your final product.

Your team has six weeks to make a conceptual design, make a detailed design, develop functional component codes, assemble component codes into a system, test the system and prepare a presentation with a live demo.

We will be using the course Blackboard for recording your team's software-design and development processes.

**Team Member and Team Name Preference (Due 10/29 (Mon) 23:59)**

You may express your preference regarding with whom you would like to team up. The group leader must send me an E-Mail with team name and the list of members by the deadline. Each team will have an area in the Blackboard where you can post and share materials and codes for the project with team members. You can also start discussing what to build.

**Conceptual design: Due 11/05 (Mon) 23:59**

***Group Meeting 1 (At least one hour between 10/29 and 11/5)***

Meet as a group and brainstorm various ideas for your software product. Choose your teams' best product idea, and give it a temporary product name, and discuss the idea further by answering the following questions:

- Product Identification
  - What are the basic functions of the product?
  - What are the special features of the product?
  - What are the performance targets?
- Market Identification
  - What is the target user group?
  - How large is the user group?
  - What are the competing products?
- System Description
  - What are the major functional components in the product?
  - (Assign each of the components to one of the team members.)
  - How should the functional components be assembled together?

Immediately after the meeting, or during the meeting, post the product name, a one-paragraph description of the product, the Product Identification, the Market Identification and the System Description, on the course Blackboard under your team's directory. In your document, clearly indicate who will be in charge of each component.

**Detailed design: Due 11/12 (Mon) 23:59**

***Individual Work (between the first and second group meetings)***

Each team member will refer to the conceptual design document and draft high-level pseudo code (although we haven't covered in class, if you prefer to use a flow chart, you can substitute a flow chart for the pseudocode) of the component(s) that he/she is in charge of. Post your pseudocode on course Blackboard under your team's directory before Group Meeting 2 – make sure to include your name in your flowchart and pseudocode.

***Group Meeting 2 (At least two hours between 11/5 and 11/12)***

Meet as a group to review and discuss the flowcharts and pseudo codes. After each member presents his/her flowcharts and pseudocode to the other team members, discuss and confirm that each member's component design fits the initial conceptual design. Modify flowcharts and pseudocode as necessary during the meeting.

In the same group meeting discuss how all the components should be assembled together to form a complete product. Discuss how all the components are interfaced to each other.

Also make sure to discuss and agree on:

- what computer platform the team will use for the final assembly of the functional components.
- who will be the assembly leader, who takes the lead in integrating all the components into a single assembly. (As this may be a challenging task, you should appoint a member who has stronger programming skills.)
- what data structures and classes should be defined and shared among the members.

Immediately after the meeting, or during the meeting, write a summary of your discussion, and post it on the Blackboard under your team's directory.

**Component code: Due 11/19 (Mon) 23:59**

***Individual Work (Start immediately after the second group meeting.)***

Each member writes the code for the component(s) that he/she is in charge of. Also write a code for debugging and test the component code. Post a 1-2 page document that describes the code that you developed and how you tested the functionality of the code. The assembly leader should keep in touch with other two members so that all the component codes will fit together in a consistent way.

**Alpha-version code: Due 11/26 (Mon) 23:59**

***Group Meeting 3 (At least two hour in the week of 11/21)***

After all the components are developed, meet as a group to assemble all the code into a single system. Work together as a group to debug and integrate all the code together.

Discuss how the two requirements, a graphical display and user interactions, can be achieved in the final product. Also decide how the functionality and the robustness of the system should be tested.

Immediately after the meeting, or during the meeting, write a summary of your discussion, and post it on Blackboard under your team's directory.

Your alpha-version code needs to be complete in terms of functionality and has to run as a unified system even though it may have some bugs. Submit your alpha-version code on the Blackboard and post a 2-3 page document to show that the system runs and achieves the intended functionality.

The document has to be posted to the blackboard under your team's directory.

**Final-version code: Due 12/03 (Mon) 23:59**

Continue testing and debugging the system code, and submit your final-version code on the Blackboard. The document has to be posted under your team's directory.

***Group Meeting 4 (At least one hour in the week of 11/26– the earlier, the better)***

Meet as a group after your final-version code is completed. Discuss how the team will make a presentation and show a demo in the final presentation. Presentation package must also be uploaded to the Blackboard under your team's directory.

**Project Presentation (12/04 and 12/06)**

Each team will be asked to make a product presentation on 12/04 and 12/06. All teams should be ready to present on 12/04. Each presentation should be 15 min in length, followed by a 1-2 min Q&A session.