

APS105 – Computer Fundamentals Project

Objective: To do something interesting.

Fall 2000

Printable Copy (PDF)

- the project is worth 50 marks, or 5% of your final grade
 - draft proposal is due in the lab on November 21, final proposal on November 22
 - final report and demonstration are to be completed before your lab period December 5
 - source code to be electronically submitted by 11:59pm December 6
 - *a project grade of **zero** will be assigned if you submit anything late, are missing TA signatures, or if you don't submit the source code*
-

1 Overview

You must produce a proposal for an interesting program, implement that proposal, and write a report on the experience. The program should be of reasonable size, at least 300 lines of code, and not more than 2000 lines, including comments. The intent is for you to write something that is roughly comparable in length and difficulty to lab 4 (Checkers), except now you must design and implement the entire project.

2 Week 0, Topic Reservation

A number of ideas for topics will be shown on the following web page:

<http://www.ecf.utoronto.ca/aps105w/Project>

You may either choose from among these ideas or produce your own idea. The choice of topic is up to you; however you should think carefully what you are trying to do and perhaps discuss the project with a TA. You must be careful that you don't do too much or too little work. Be reasonable and stay within your limits. It is often very difficult to judge the amount of work required for a particular idea, so a TA will help you make this judgement.

- *To come.* There will be a reservation system to ensure that a wide variety of projects is selected. This system will be active at 3am on November 18.

3 Week 1, Proposal and Review — 10 Marks

For your first project lab period on November 21, you must submit a written proposal that outlines what you plan to do. The proposal should have sufficient detail to identify the problem *and* show that you know how to solve it. It should look very similar to any of the lab writeups you have been given so far, except that you can give away the answers too. It may include pseudocode, API specifications, descriptions of the main data structures used (arrays, linked lists, objects), diagrams, flowcharts, or drawings. Up to two marks each will be awarded for:

- (C) clarity – does the TA understand what you are trying to do?
- (D) detail – is there enough detail for the TA to visualize the solution right away, or is it too sketchy?
- (E) English – are there any spelling or grammatical errors? is your writing clear and succinct?
- (N) neatness – are the text, code and figures laid out well? is your writing/printing legible?
- (T) thoroughness – is it complete? did you leave important things unspecified? did you forget to describe important cases, rules, or program behaviour? is your proposal free of ambiguity?

This proposal *must* be on the form provided, using only one side of a single page. You may print this form once, then feed it through your printer again at home or *very neatly* handwrite your proposal.

- Bring 2 copies of the Project Proposal form to your lab period on November 21:
 1. One copy will contain a *draft copy* of your proposal. Clearly label it as being a *draft*.
 2. The other copy will be *blank*, and is to be signed and used for the *final copy*.
- Discuss your project with a TA and have him or her *sign both* forms.
- The TA will review your proposal, giving you feedback to improve it.
- After considering the TA's feedback, rewrite your proposal onto the signed, blank form.
- Staple your *final copy* on top of your *draft copy* to create your *completed proposal*. If your draft was perfect, label it as *final* and neatly strike out the draft label.
- Place your *completed proposal* in the *submission box* immediately inside the office door of SF2002. Make a copy for yourself.
- The deadline is 4:00pm on Wednesday, November 22, 2000. The office is closed 12-1pm for lunch.
- **Late submissions or projects missing a TA signature on the proposal will get a grade of zero.**
- The TA that signs your proposal will not necessarily be the one to mark your project. A randomly assigned TA may be marking your project later.

4 Week 2, Checkpoint – 0 Marks

During the November 28 lab, you should have made significant progress on your project. The TAs will be available to answer questions and help you with technical problems. You should attend your own scheduled lab time, but you may also attend the other lab times if you need a lot of help (10–noon, 2–4pm, 5–7pm).

Use the principles of software engineering (re-read Lab 4) to help you make forward progress. It is **far better** to demonstrate a simplified, working program than to demonstrate a complex one that does not even compile. You may have to begin cutting back on your plans if you don't think you will complete your project in time.

5 Week 3, D-day – 20 Marks

December 5 is Demo Day, so you should be prepared to demonstrate your working project at any time during your lab.

You will get up to 5 marks (maximum of 20) for each of the following categories:

- (C) compiles – your program compiles error-free, assuming there are at least 200 lines of useful code (eg, they are not all comments or print statements)
- (R) runs – you can show that it runs correctly on simplified input or on a simplified problem
- (T) thoroughness – you can show that it runs and does everything correctly
- (O) outstanding – your program is truly outstanding and well beyond what was expected

These category marks work like a thermometer, in that you must get *all* 5 marks in the previous category to be eligible to get any marks in the next one. For example, you must get all 5 R marks to get any T marks. However, missing one or two small corner cases won't prevent you from getting all 5 R marks – this category is intended to help students who have trouble getting their whole project working.

The TA may assign the O (oustanding) marks to your project after the lab is over.

After your demo, you should submit your source code. You may make small bug fixes and improvements, but these changes will not influence your demo grade (it will merely allow the report marking TA to run your program easier). The deadline for submission is 11:59pm Wednesday December 6. You must submit *all* files required to compile and run your program. You should also submit any special instructions (eg: how to run your program — using java, appletviewer or a web browser, etc) in a file named 'README'. For example:

```
computer.ecf% submitaps105f 9 README Project1.java Project2.java picture.jpg Project.html
```

6 Final Report – 20 Marks

You are expected to write a final report based on your experience. You should describe what you planned to do, what you did and why it was interesting, worthwhile, and/or provided a good learning experience.

On the front of your report, you must place a standard Grading Form you will be given. The Grading Form must be clearly visible when your report is closed. For example, you may use a clear, transparent report cover or a simple staple to bind the report.

<http://www.ecf.utoronto.ca/aps105w/labs/project/grading.pdf>

The total length of your report should be no more than 1500 words. Include any tables, diagrams and pseudocode needed to illustrate your work (do not include them in the word count). Shorter is OK, as long as you feel you've said what needs to be said. Your report should have five sections and an appendix, as described below:

1. **Cover Page.** The Grading Form, followed by your completed proposal. The TA will return your proposal during the demo.
2. **Introduction and Motivation.** What you're doing and why it is interesting, worthwhile, and/or provides a good learning experience.
3. **Design.** Describe how your programs works, the major methods and their goal, and how they interact.
4. **Shortcomings.** Describe any known bugs, errors, or omissions in your program. Like all scientific work, it is your duty to be honest. You will not have marks deducted for honesty.
5. **Conclusions.** Summarize your project in a few words, and say what you would do differently if you were going to do it all over again.
6. **Appendix.** A printed copy of your code, including comments.

While marking your report, the TA marking your project may find it necessary to adjust your proposal and demo grades (either up or down). This is done to ensure fairness in marking and catch things that may have been missed by the other TAs. The report will be marked out of 20, with four marks each for:

- (C) clarity – does the TA understand everything that you did?
- (D) design – did you design your program well? did you use loops where appropriate? did you use methods where appropriate?
- (E) English – are there any spelling or grammatical errors? is your writing clear and succinct?
- (N) neatness – are the text, code and figures laid out well? is your writing/printing legible?
- (T) thoroughness – is the report complete or too short? did you leave important things out? did you forget to describe important cases, rules, or program behaviour? did you make errors in your project that aren't listed in the shortcomings?