

Final Project

Please read the following document carefully. It contains several important guidelines.

Checkpoints for final project

DATE	WHAT IS DUE (discuss with instructor or TAs)
Fri Apr. 8	1-2 page proposal writeup including intended example implementation and rough class hierarchy
Tues Apr. 19	Progress report (meet with instructor)
Mon-Tues Apr. 25-26	Final project demo and poster showing (Lab sessions)
Wed Apr. 27, 5 PM	Final paper due

General Guidelines

You will work on groups of two. You will need to document the individual contributions in your README file showing your work.

Your project should follow the programming guidelines, and make use of *inheritance*, *Standard Library*, *exceptions* for error handling, and implement a *GUI*. The GUI must be in Gtkmm. If you want an exception to choose some other GUI, you need to have prior experience. Specify what projects you have undertaken in your proposed GUI toolkit in your proposal. Be prepared to show these to the instructors. You will be required to write a paper, prepare a demo for the last week of lab, create a PowerPoint poster which you will show in the demo, and finally make a CD of everything for archives.

The recommended final project is to implement the document editor from the design patterns hand-out, using Gtkmm, inheritance, and the standard libraries. Other projects should have at least the same complexity, and will be evaluated in your proposal.

Proposal and Progress Report Guidelines

The proposal is due **Friday, April 8**. It should specify the following within a writeup (not a list):

- Who is working together
- Proposed title of your project
- A description of your proposed project
- A rough sketch of how you propose to implement your project, including a draft of the class hierarchy and your proposed GUI toolkit if it is different from Gtkmm.
- An outline of what parts of the project each group member will undertake.

A progress report will be due **Tuesday, April 19** and you will have a meeting with the instructor that week. This report should reflect progress you have made, and should reflect back to the proposal.

Paper and Presentation Guidelines

- The **literature survey** should include at least 3 references to books and articles. It will be necessary to present the papers and books (or copies thereof) that you will be using. At least one of them has to be to a scientific journal, and one should be different than the ones provided by the instructor (if any). You may use the following resources to help you with the bibliographic search:
 - ACM digital library: <http://www.acm.org/dl> You may register and create a non member account. The search engine is very powerful and allows you to create binders on different subjects. You may find the articles by looking them up in the library, and if you do not succeed you may purchase it online. If you consult with the instructor, you may be reimbursed for the article purchase.
 - IEEE Computer society online:
<http://www.computer.org/publications/dlib/index.htm> Access here is more restrictive, but you will be able to find titles and abstracts.
- The **conceptual definition of example** will describe an example that illustrates the general area of research for the final project.
- The **demo** will be performed in the Monday and Tuesday lab sessions of the final week of classes. Each group will choose timeslots either on Monday 4/25 between 3-4:30 or Tuesday 4/26 between 3:30-5. You will demo your code on the computers in the lab, either using Visual Studio if it is Windows compliant, or with SSH and X-Win32 if it is Linux compliant. In addition, you will show a **PowerPoint poster** describing your contributions.
- Each group should also produce a **CD** of all code, executables and project files for the archives, and turn this in at the demo.
- Regarding the **final paper**, keep in mind the following points:
 1. It should contain an *abstract*, a paragraph describing the problem and your contribution. The abstract is separate from the rest of the paper.
 2. A list of *keywords* should follow. These are the main concepts (not necessarily one word each) explored in the paper.
 3. The *introduction* should describe the problem addressed and the general solution strategy as succinctly as possible. Repeating the abstract here is acceptable.
 4. It is important that it includes *references to the literature (and relevant web sites)*. It is not enough to cite a paper, you should describe how it relates to your problem and to your solution (which means you need to read at least the abstract or introduction, and the conclusions of each reference you give).
 5. The *main body* should make a logical presentation that supports the claims you make in the abstract and introduction. Supporting figures, tables, diagrams, etc., should be here. It should be dispensable (that is, people who only read abstract, introduction, and conclusions should realize why it would be interesting to read all of your paper).
 6. The *summary or conclusions* should really follow from the introduction and main body. Do not make wild conclusions that do not follow from the rest of your paper, or introduce something totally unexpected (i.e., you are not writing a suspense story). Speculation should be included in a *future work* section.
 7. The *references* section should follow.
 8. You may include **appendices** if you want to show code segments or program diagrams, more detailed results which do not belong in the main exposition, etc.
 9. The proper *formatting of code* is to use typewriter font for **code segment**.

10. If you quote literally from some sources, indicate so by placing the text in between quotation marks, and using common English usage rules of style. Do not copy too many or too long sections. If you are paraphrasing, cite by saying “cf. such-and-such-citation”. You will not be granted extra points for verbose papers.
 11. Finally, you should include your no-more-than 75 words *biographical information* and a *digital picture*.
- Please produce your write-up using the word processing system **LaTeX** and print it on a laser printer, including computer-generated graphics. Macros for LaTeX are downloadable from <ftp://ftp.springer.de/pub/tex/latex/lncse/> You will need the files with suffixes cls, doc, and the read.me file. Your TAs will give you more pointers on LaTeX tutorials and examples. This is not an arbitrary imposition: it will help produce **high quality proceedings**, with a table of contents. You will receive a copy of these proceedings, and will be able to share them with friends and family.
 - The final paper for grading will be due on **Wednesday, April 27 at 5 PM**. We may make corrections and submit it back to you for revisions.
 - Please take a look at the following papers in the CSE331 (data structures) online final project proceedings, for examples of outstanding projects and papers in all respects, at <http://www.nd.edu/~cseprog>:
 - Nick Petroni and Aaron Walters, “Overcoming the Insecure Medium via Encryption,” (1999)
 - Brian Barrett, Lisa Pellegrino, and George Viamontes, “AtomAnalyze: A GGCL Implementation,” (1999)
 - Tim Dysart, Andy Sordi and Joe Taylor, “Image Compression Using Quadrees”, pp. 44-57 (2000)
 - Branden Moore and Randy Rausch, “Notre Dame PalmOS Music Player”, pp. 91-98 (2000)
 - Jim Maher, Andy Janiszewski, Steve Schrantz and Will Bruckert, “Multi-Player Checkers: Glade Generated GUI & Socket Communication”, pp. 99-105 (2000)
 - Maureen Hillenmeyer and Margaret Lewis, “cDNA Screening Tool” (2001)
 - Craig Martus and Joe Lammersfeld, “Zelda: The Next Saga” (2001)
 - Mudit Agarwal and Ryan Kennedy, “Blackjack” (2002)