

# Term Project Assignment

## 1 Schedule

Sunday, November 13	One-page project proposal due.
<b>Friday, December 9</b> (last day of Reading Period)	<b>Project and report due.</b> This deadline can be extended only by a Temporary Incomplete authorized by the student's residential college dean.

## 2 Requirements

The purpose of the project is to give you a chance to explore in some depth a topic that is related to the subject matter of this course. The project consists of two parts, an *activity* and a *report*.

1. The activity involves scholarly work—exploring a topic, reading relevant literature, writing software and running experiments, and so forth—and must include a substantial amount of programming.
2. The report is a short paper (8–15 pages) describing the results of the activity and should reflect the substantial amount of work that you put into the activity. The paper should make clear how you approached your topic, what resources you used, what you actually did (papers read, code written), and what you learned. It should present your own conclusions and perspectives that you acquired in the course of your work. The report will be graded for writing quality as well as for technical content, so attention should be paid to organization, grammar, spelling, and scholarly style. All references used, both text and code, must be properly cited in the bibliography.

### 2.1 Project Proposal

The project proposal should state what you intend to do and to briefly outline how you plan to carry it out. It should identify needed reference materials and software tools. The purpose is to make sure you get started early on the project and to enable me to provide early feedback. A PDF of your one-page project proposal should be submitted online as assignment number “8-proposal” using the Zoo `submit` script.

### 2.2 Final Project Submission

The completed project, including the project report, code written, and any supporting materials such as experimental data collected should be submitted online as assignment number “9-project” using the Zoo `submit` script.

### 3 Project Suggestions

Almost anything related to the course material is acceptable as a project topic. Because this is a course on *object-oriented programming* the main criterion for appropriateness is work that relates to the design and implementation of object-oriented programs.

One kind of appropriate project is to develop a substantial application, where the project is to specify, design, implement, and document a C++ program to solve a particular task such as playing a simple game, maintaining a CD-sharing library, or handling play lists in a music player.

A word of warning here: Time is short, so the chosen project must be quite limited in scope. People often vastly underestimate the amount of time (and code) required to solve a problem. A small project that is carefully done and documented is much better than an overly-ambitious one that is half-finished. Anyone proposing a development project should structure the proposal into a sequence of steps, where the first step provides just a skeleton of the full application, and successive steps along the way add features a few at a time. That way, when time runs out you can wrap up the portion completed and have something to show for your efforts.

Another kind of project is to explore a topic related to the course in greater depth. For example, one might study design patterns, compare object-oriented design with other popular software design methodologies, or investigate large C++ template libraries such as STL, Boost, or gtkmm, with an eye to learning how they are designed and implemented.

Performance studies might also make good projects, e.g., explore some of the available profiling tools in order to get better understanding of where a program spends its time than is possible with the simple stopwatch tool, get a better understanding of the cost of polymorphic derivation, or find out how good the implementations are of some of the algorithms in the standard libraries.

Much more important than the choice of topic is what you are able to do with it. I want to see an original well-written report that explains, integrates, and contrasts material from various sources, not one that simply condenses or paraphrases. If reporting on a development project, then the report should describe the goals for the application, the program itself, its strengths and limitations, and how the design methodology learned in this course was used in creating and debugging the code. For a research-style project, the paper should state the problem, survey the relevant literature and prior work, explain what you did, and present your results and conclusions.

My goals for the project are that you should learn something new, your paper should teach me something new, and you should write a substantial amount of high-quality C++ code involving several classes.

**For undergraduates registered for CPSC 427a:** I expect a well-written report that reflects a significant amount of effort put into the project.

**For graduate students registered for CPSC 527a:** In addition, I expect the project and report to show a greater degree of originality and professionalism than I expect from undergraduates.

## 4 The Report

### 4.1 What Makes a Good Essay?

Past experience has shown that many Yale students do not know how to write an essay or scholarly paper. I can't really tell you how to write a paper in this brief handout, but the section title is intended to remind you that a project report *is* an essay of sorts, and the things you have learned in other courses about logical organization, writing style, use of proper grammar and spelling, and proper methods of citing other people's work all apply here.

Like an essay, the project report should have some ideas of your own to report. It should reach some conclusion, and it should give logical arguments and relevant data to support that conclusion. What I do *not* want in a paper is a simple paraphrasing of somebody else's work. Quoting other people's work as a way to make a point is perfectly acceptable, if properly attributed; simply copying their work, attributed or not, is not acceptable. I want to read about *your* ideas, *your* code, and *your* conclusions, not somebody else's. But of course you will rely on other people's work to support your arguments.

### 4.2 Required Form of the Report

Your paper should follow accepted guidelines for scholarly work in computer science. The paper should begin with title, author, date, and abstract. Every page must be numbered. The body of the paper should be divided into logical sections appropriate to the structure of the material. Each section should have a numbered section heading. Numbered and unnumbered subheadings should be used where appropriate. Figures and tables should have captions and should be referenced by number. Related work should be cited in the text, and full reference information should appear at the end in a bibliography. Any material copied verbatim should be enclosed in quotation marks as well as being properly cited. The bibliography should contain full citation information, including author, title, year, and publication data. If the publication is a conference proceedings, then the proceedings title, editor, organization or publisher, etc. should be included. If it's a book, then the publisher should be included. If it's a web page, then the URL and sponsoring organization should be mentioned.

### 4.3 Writing Tools

Most research papers in computer science are prepared using the  $\text{\LaTeX}$  typesetting system. (That is also the system that I use for preparing class handouts!) Because this is a computer science course, and because it is educational for you to learn to use the tools of the field, *you are required* to use  $\text{\LaTeX}$  to prepare your paper and to use the companion tool,  $\text{\BibTeX}$ , to prepare your list of references. *Exceptions require my prior approval.*

Compared to papers produced on a typical word processor, the results from  $\text{\LaTeX}$  are much more professional looking.  $\text{\LaTeX}$  numbers pages and sections, automatically produces a table of contents and list of figures (if desired) and generates page headers and cross-references.  $\text{\BibTeX}$  takes information that you put into a bibliographic database and formats it according to commonly-accepted styles. You don't have to remember whether the article title or journal title should be italicized, or where commas and periods are needed—it does all that for you. But the big win comes when typesetting equations and other mathematical notation.  $\text{\LaTeX}$  produces professional-quality typeset equations and

formulas rather painlessly. Attempting to do the same in a word processor is clumsy at best, and the results are generally disappointing.

It does take some effort to learn how to use  $\text{\LaTeX}$  effectively. I would suggest looking at one of the two tutorials [Gre00, OPHS08] that are accessible via the “Online Documentation” section of the course web page. For the advanced  $\text{\LaTeX}$  user, I recommend [Lam94, GMS94]. There is also a wealth of information on the CTAN archives, available on the web at <http://www.tug.org>.

The finished report should be submitted in the form of a PDF file.  $\text{\LaTeX}()$  users can produce a PDF file directly by using the `pdflatex` command. Users of other word processors can create PDF by “printing” the file to the PDF device. The source file(s) for the report should also be submitted.

$\text{\LaTeX}$  and  $\text{\BibTeX}$  are already installed on the Zoo. Mac users should download TeXShop, available from <http://pages.uoregon.edu/koch/texshop/>. Windows users should consider MiKTeX, available from <http://miktex.org/>.

## 5 Hints

*Start now!* Locate your resources. Many but not all papers are available on the web. Non-electronic library materials can sometimes be hard to get and may need to be obtained on interlibrary loan (which can take weeks). Make sure your project is doable in the amount of time available. Your proposal should identify the resources necessary to carry out your project.

The Yale library subscribes to many on-line publications and makes them available to the Yale community without charge. However, they must be accessed from a Yale computer and sometimes through the library web site. The particular access method varies from journal to journal. Help is available from

Andy Shimp  
 Engineering & Applied Science Librarian  
 Yale University  
 15 Prospect St./P.O. Box 208284  
 New Haven, CT 06520-8284  
 Tel.: 203-432-7460  
 Fax: 203-432-7465  
[andy.shimp@yale.edu](mailto:andy.shimp@yale.edu)

If you are not on campus, you can still access Yale Library materials by using a VPN or proxy server and your Yale NetID to connect to the Yale network. See <http://library.yale.edu/about/offcampus.html> and <http://www.yale.edu/its/network/vpn.html> for further information.

If you’re doing a development project, you might find that you need software resources that are not currently installed on the Zoo. If so, please let me know as soon as possible what you need and I will do what I can to get them installed quickly.

## References

- [GMS94] Michael Goossens, Frank Mittelbach, and Alexander Samarin. *The  $\text{\LaTeX}$  Companion*. Addison Wesley Longman, Inc., Reading, Massachusetts, 1994.

- [Gre00] Harvey J. Greenberg. A simplified introduction to  $\text{\LaTeX}$ . Available for download from <http://www.ctan.org/tex-archive/info/simplified-latex/>, April 2000.
- [Lam94] Leslie Lamport. *A Document Preparation System:  $\text{\LaTeX}$  User's Guide and Reference Manual*. Addison Wesley Longman, Inc., Reading, Massachusetts, second edition, 1994.
- [OPHS08] Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl. The not so short introduction to  $\text{\LaTeX}2\epsilon$ . Available for download from <http://tobi.oetiker.ch/lshort/lshort-letter.pdf>, September 2008. Version 4.26.