

Instructions for Submitting Labs

Distributed: Monday, October 12th.

A complete lab submission includes all of the following parts:

1. Documentation
2. Implementation
3. Peer Evaluation (by each team member)
4. Proposed Grader Tests

1 Documentation

You should produce “professional-looking” documentation: documentation that you would be proud to show your boss. Use a good word processor (*e.g.*, LaTeX, Word, Pages, Writer) to facilitate document preparation.

Your documents should be well written, logically structured, and pleasingly formatted. Hint: leave time for other members of your group to proof read what you write, and vice versa since your grade hinges partially on the rest of your group. This applies to both native and non-native English speakers. Give each other advice on their writing style. Leave time to “smooth over” the writing: the documentation should flow as if all of it was written by the same person, even though not all of the documentation is tailored to the same audience.

The organization and style of your writeup are almost as important as its content. If a prospective user cannot find necessary information about your program, he or she is likely to give up on your program and look for another. Above all else, you should be concise. Try to avoid redundancy as well as ambiguity and omissions. To convey relationships among elements of your report, use tables and pictures rather than prose whenever possible.

You should have your documentation bound into a single volume (*e.g.*, with a three ring binder). Each part is separated by a divider, making it easy to open to any particular one.

1. Front Matter
2. User’s Guide
3. Programmer’s Guide¹

¹In *addition* to the hard-copy version, it might be appropriate to submit some parts of the programmer’s guide electronically as well.

4. Test Plan
5. Meeting Minutes
6. Appendices
7. References
8. Errata Page

The front matter consists of a title page, table of contents, and introduction. The title page should include at least the title of the project, the names of the group members, the group name, and the date of submission. The table of contents should be fairly comprehensive and should list all the sections and the subsections of the report in the order in which they appear and the page numbers on which each of them begins. It should be well-designed and should distinguish between sections and subsections by using upper/lower case letters and indentations. Figures and tables should be listed separately after the contents. The introduction gives a general overview of the project. It should provide the concepts on which the project is based and how it works. It should also lay the foundation for the other sections.

The four principal components (User's Guide, Programmer's Guide, Test Plan, and Meeting Minutes) are self-contained documents in their own right. Each will have its own title page, table of contents, introduction, etc. Separate handouts describe the recommended contents of each of these documents.

If you are using Javadoc to write (parts of) the Programmer's Guide, you will likely find it easier to create electronic html-based documentation rather than a hard copy. Doxygen can be easily used to produce nice hard copy printouts, but Javadoc is a little less flexible. If your Programmer's Guide contains parts that are html-based, submit that document (ie that web site) electronically using the CSE submit command. You will want to submit an entire directory, named something descriptive like "documentation", with a README file that indicates what file is the main (ie root) page. In *addition*, please turn in printouts of those web pages as part of your hard-copy Programmer's Guide. These printouts are for feedback purposes only. That is, it is understood that the format and structure of this part of the documentation comes from the electronic submission. The hard copy is only for our convenience, so we can mark up your submission and record our comments with an old-fashioned pen.

The appendices contain information that, while potentially useful to the reader, can be easily omitted from the main document without loss of critical information.

The reference section gives citation information for any book, articles, journals, magazines, or technical documents referred to in the text of the documentation.

Finally, the errata sheet describes all the known bugs and limitations of your implementation. Errors described in the errata sheet will be penalized *less* than errors discovered by the grader!

Submission and deadline. The documentation is turned in at the beginning of class on the due date of the lab.

2 Implementation

Your implementation will be submitted using the CSE submit command. You should submit the code, not a compiled executable. Make sure to also include a README file with specific instructions on how to compile, install, and run your submitted project. Ideally, you would also include a makefile to simplify the compilation and installation of your submission.

You can tar your implementation files into a single file, then submit the whole collection with (for example):

```
$ submit c560ab lab1 sim.tar
```

If you are using a platform other than the CSE cluster (ie Unix, Windows XP), you must turn in a floppy disk or cd-rom with your documentation. The grader will install your implementation from this disk during your interactive grading. (You should bring a laptop with the appropriate operating system to your interactive grading.)

Submission and deadline. The implementation must be submitted by the beginning of class on the due date of the lab.

Optional early submission. If you wish, you may submit a *preliminary* version of your implementation. This version is for the graders to use to debug their own test cases. *It will not be graded.* If you submit an early version, please do so by noon on the day **before** the due date of the lab. Send email to the graders to let them know that your implementation is available to them. Don't forget to submit your final version before the usual deadline.

Early submission allows the graders to check that their test cases meet some basic requirements expected by your implementation. This is to your advantage since, otherwise, graders may have to spend time *during your interactive grading time window* modifying their test cases.

3 Peer Evaluation

Each student must submit their *own* peer evaluation form. The peer evaluations are kept confidential, so you will not know what grades your team-mates have given you, or vice versa. The peer evaluation form includes a quantitative evaluation of each member of the group, including one's self). These peer evaluations form the basis of the individual grade component of the lab grade (10%). **Note: if you fail to submit a peer evaluation, you will receive 0 for the individual portion of your lab grade.**

If everyone pitched in reasonably, then everyone should get an 8 (reserve a 10 for truly exceptional work). **The total of the individual ratings must not exceed $8n$, where n is the number of people in your group.** Include comments to justify and clarify the ratings you have assigned.

Submission and deadline. The form is available as a survey on Carmen. The survey must be completed by 5pm on the due date of the lab.

4 Proposed Grader Tests

The proposed grader tests are a suite of no more than 3 test cases (properly documented) to be included in the grader's test suite. You should choose some test cases that you feel exercise some particularly subtle or difficult aspects of the lab. Describe what each test case is meant to accomplish (ie what feature or aspect of the specification is being tested) as well as the expected output.

Submission and deadline. The test cases must be sent by email *to the graders and instructor* by noon on the day **before** the due date of the lab.