

OpenGrade

Benjamin Crowell

www.lightandmatter.com

Contents

Introduction	1
Installing	1
Using the Graphical User Interface	2
Goodies	5
Using the Text-Based User Interface	5
Safety Features	6
Passwords and Authentication	6
Posting Grades on the Web	7
How Grades Are Calculated	8
Format of Gradebook Files	9
Future Improvements	10
Bugs	11
Programming Stuff	13
Licenses	13

1 Introduction

- purpose** OpenGrade is software for teachers to keep track of grades. It can put the students' grade reports on a web server, and allow the students password-protected access to them.
- legalities** OpenGrade is free software, and it comes with source code. It is copyright 2002 by Benjamin Crowell, and to have permission to copy it, you must agree to the terms of the licensing agreement on page 13. This documentation is copyright 2002 by Benjamin Crowell, and is available under the GFDL 1.1 license on page 19.

2 Installing

- Linux** You will need a Perl interpreter, which is available as free software. I'm using Perl 5.6, but I think 5.0 will probably work. You will also need the free gcc compiler installed.

First, download the `opengrade.tar.gz` file from `lightandmatter.com`. If your browser doesn't automatically unpack the file, you can unpack it by using the Unix command `tar -zxvf opengrade.tar.gz`. Move the resulting directory tree to wherever you want it to live permanently.

Get in a terminal window, and move into the directory you just created. Next, you'll run an installer script. This script will download some open-source software from `cpan.org` that is required in order for OpenGrade to work. (The modules it uses are `Term::ReadKey`, `Date::Calc`, `Net::FTP`, and `Tk`, used by the GUI, not needed if you only use the text interface.) The installer script will check whether each of these is already installed, and if it's not, it will

download and install them. You need to have an active internet connection for this to work. The command to run the installer is

```
su -c "make install"
```

For convenience, you'll want to set things up so you can run OpenGrade from any directory. On my own system, for instance, I have a `~/bin` directory that's in my path, and in that directory I have a symbolic link that points to OpenGrade in the directory where I have OpenGrade's source code.

After all this, you can run OpenGrade by invoking the program from the command line. You can run the terminal-based interface like this:

```
% OpenGrade -t
```

Once you're in the program, you can open or create a file. If you like, you can supply the name of the file on the command line,

```
% OpenGrade -t my.gb
```

The `.gb` extension stands for gradebook. If you omit the `-t` switch, you'll run the GUI:

```
% OpenGrade &
```

or

```
% OpenGrade my.gb &
```

MacOS X The text-based user interface runs fine on MacOS X — that's the system I developed it on. The GUI requires Perl/Tk, which is currently (as of April 2002) not easy to install in MacOS X. You need to install the Developer Tools in order to get the C compiler installed. They come either on a CD or in a `.dmg` file cleverly hidden somewhere on your hard disk. Since MacOS X is Unix, the installation procedure is similar to the Linux procedure. The command to run the installer is

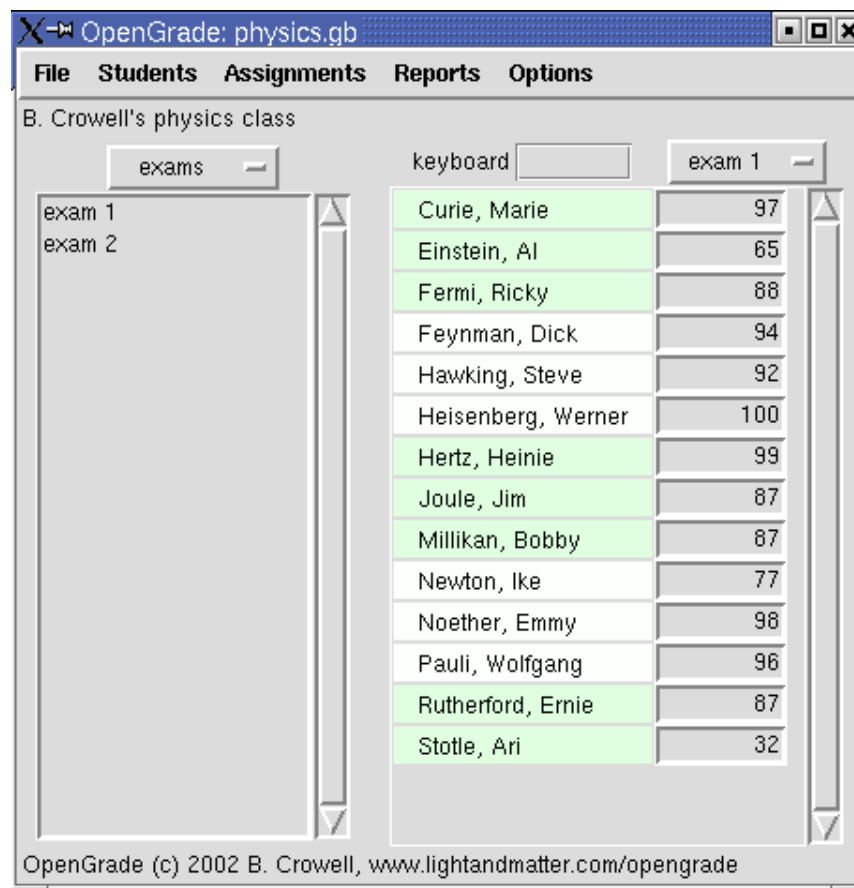
```
sudo make install
```

Windows First you need to download Perl from <http://www.activestate.com/>. Starting from their homepage, follow the link for downloading ActivePerl. The first place this takes you is a page that helps you download any special software you need for your particular version of Windows. (If you aren't sure what version of Windows you're running, you can find out by right-clicking on My Computer, and then selecting Properties.) Next they ask you for some registration information, and then you download ActivePerl.

Once you have ActivePerl installed, download OpenGrade, and unpack the archive using WinZip or similar software. Reboot, make sure you have an active internet connection, and then double-click on the file `install_win.pl`. Wait until it's done, and reboot again. To run OpenGrade, double-click on the file `ogr`.

3 Using the Graphical User Interface

entering grades To see how OpenGrade works in everyday use, run the program as described above, and open the file `sample.gb` that was included among the files you downloaded. Enter `secret` for the password. (OpenGrade uses passwords for authentication, but there is no encryption. See page 6 for more information on how this works.) Click on exam 1 in the list on the left, and you should see something like the window shown below.



On the right, you have a list of the students enrolled in the class. Their scores on exam 1 have already been entered, but you can change them now if you like. If you click on exam 2 from the list, you'll see their results from that exam.

Note the popup menu at the top of the column of scores. With this menu, you can choose to see each student's average test grade, or each student's overall grade in the course. The two tests are the only assignments in the file right now, so the exam averages are the same as the overall grades.

If you explore the other popup menu, above the list of assignments, you'll see that there is another category, homework, with no assignments in it. Let's add a new homework assignment. Select the homework category from the popup menu, and then select New Homework from the Assignments menu. Type 1 in the first space: this assignment will be known as Homework 1. Type a number in the second space for the maximum number of points possible on this assignment. Click on OK. A blank column of scores pops up in the main window, and you can enter the students' grades.

When entering grades, make sure you have your Num Lock turned on, or else OpenGrade won't respond to numbers from the keypad. When you hit the Return key (on either the keypad or the main keyboard), the grade will be entered, and the next student will automatically be selected. (You can also use the down-arrow key as a synonym for Return, and the up-arrow to select the preceding student.) This allows you to enter a list of grades quickly if you already have them in alphabetical order.

If you have a stack of papers that is not in alphabetical order, and you're entering all the grades, the quickest way to do it is to use the keyboard to select each stu-

dent. Here's how it works. Normally when you're entering grades, the keyboard focus is in one of the little grade boxes on the right-hand column of the spreadsheet. If you hit the tab key, however, you'll see that now the student's name is highlighted rather than the student's grade. Once the focus is in the left-hand (names) column of the spreadsheet, you can use the keyboard to select a different student by typing the first letter (or first few letters) of her last name. The characters you've typed will appear above the top of the list. Now hit the tab key again, and you can enter the grade. By hitting the tab key, you can cycle back and forth between selecting students and entering grades for them. Note that if you keep on typing, the software assumes you're trying to be more specific about what student you have in mind rather than selecting a different student. For instance, if you have students named Jarret, Johnson, and Olson, then typing jo means that you're selecting Johnson; it doesn't mean you hit j, then changed your mind and wanted to select Olson instead. If you hit j and then change your mind and want to select Olson instead, then you need to either hit the backspace key to erase the j. (The software will also forget about the letters you've typed when you hit tab and take the keyboard focus away from the list of students.)

creating a new file To create a new gradebook file, choose New from the File menu and fill in the form. You've created a new gradebook file, which is empty. Now you need to get it set up correctly.

If you use a fixed set of grading standards, you can now enter them by choosing Grading Standards from the Options menu.

Next you'll probably want to set up all your categories of assignments, such as exams, homework, and quizzes. To do this, choose New Category from the Assignments menu. The first thing you're asked for is a short name for the category. It doesn't matter too much what you pick; in the graphical user interface, you'll never even see it again. However, you will see this short name if you ever use the text-based user interface, or if you examine the gradebook file directly in an editor. Just pick something short, like e for exams, or q for quizzes. Make sure you don't use the same short name for two different categories. Most of the rest of the inputs are fairly self-explanatory, except for the space at the bottom for the weight of the category. This is optional. If your student's grades are simply going to be based on the sum of their points in the various categories, then you should always leave this space blank. On the other hand, if you want each category to have a certain weight, then you should enter a weight for every category you create. See page 8 for more information on weighted grading.

Once you've created your categories, you can enter your students' names using Add from the Students menu. If you already have your roster in a text file, you may find it more convenient simply to open your gradebook file in a text editor, paste in your roster, and then put it in the form OpenGrade requires, as documented in section 10.

reports A variety of reports is available from the Reports menu. The reports offered in this menu will depend on what student, category, and assignment you've selected. When you choose a report from the menu, it's displayed on the screen. You can then hit OK if all you wanted to do was look at it, or Save to save it to disk as a plain text file, e.g. for printing.

4 Goodies

This section documents various features of OpenGrade that you may or may not need. Almost everything in this section applies equally to the GUI and the text-based interface.

- extra credit If an assignment is going to be for extra credit, you can simply set it up with a maximum point value of zero, but enter scores that are greater than zero. Normally, the software beeps at you and asks for confirmation when you try to enter a grade that is higher than the nominal maximum for the assignment. But when the maximum is zero, it will accept nonzero values without complaining.

Sometimes you have an assignment that is not completely extra credit, but it may be possible for students to do something extra that gives them more than the nominal maximum number of points. When you enter a score that's above the maximum, the software will pop up a dialog box asking you if that's what you really want. To avoid this dialog box, you can just type in the score with a trailing letter x, e.g. 101x on an assignment with a nominal maximum score of 100.
- dropping a student When you drop a student, the student's grades will still be in the gradebook file, so that if you need to, you can add her back in later. In the GUI, you drop a student by clicking on her name in the main window, and then selecting Drop from the Students menu; to reinstate her, select Reinstate from the same menu.
- scores that don't count You may wish to record scores that don't count at all in a student's grade. (For instance, you might give a diagnostic test, or record whether the student has completed some safety training.) There is an option for this when you set up the category.
- dropping some scores When you're setting up categories, the software will ask you how many scores to drop in this category. For instance, you could choose not to count each student's two lowest homework scores.

5 Using the Text-Based User Interface

The user interface is menu-driven, and is fairly self-explanatory.

Often you'll need to select a particular student. Typically you'll just enter the first few characters of the student's last name, and hit return. As you go along, the software will display a list of all students whose names match the characters you've typed so far. When you see that there's only one student on the list, then you know you're ready to hit return to select that student. Capitalization is ignored here.

There is a time-saving trick that you can use when you have more than one student with the same last name. For instance, my school has a large Vietnamese population, and my classes usually have several students named Nguyen. Suppose my class includes both Phong Nguyen and Thu Nguyen. To select Phong Nguyen, I can type ng, which narrows it down to these two students, then hit the tab key, and then type p.

6 Safety Features

Grades are valuable information, and OpenGrade has a whole repertoire of methods to help you make sure you never lose any of them. All of these safety features are enabled by default, and can be turned on or off by editing the preferences section of the gradebook file using a text editor.

- auto-save Let's say you're editing a gradebook file called `foo.gb`. Periodically, OpenGrade will save a copy of your work under the filename `#foo.gb#` (the same naming convention as the one used by the Emacs editor). If your computer crashes in the middle of an intense grade-recording session, you should be able to recover most of your work from this file. OpenGrade will also attempt to auto-save your file when you terminate the program abnormally, e.g. by typing control-C. The auto-save file is deleted when you eventually save your file in the normal way, so your directory should not get cluttered up with auto-save files.
- auto-backup When you save the file `foo.gb`, OpenGrade will preserve the old version of the file under the name `foo.gb~` (again, the name is based on the Emacs convention). This takes care of the situation where you save your file, and then realize you've messed something up.
- server backups Whenever you upload your grades to your web server to allow students to check their grades, OpenGrade will save a copy of your gradebook file there under a filename like `foo.gb-2002-03-15`.

7 Passwords and Authentication

Every OpenGrade file has a digital watermark that is based on the password you choose for the file. Suppose one of your students somehow gets access to your gradebook files. Of course, this shouldn't happen, because you should store your gradebook files on a computer account that nobody has access to without logging in, but let's assume that you got up from your desk without logging out, and a student has walked up and is trying to tamper with your files. If she tries to open one of your gradebook files using OpenGrade, it will demand the password, and will refuse to open it without it. (But don't leave your desk while you have a file open!) If she opens the file using a text editor, she could start messing around with its contents. However, when you get back to your computer and open the file in OpenGrade, it will detect the fact that the digital watermark is no longer valid. OpenGrade does not store your password anywhere, so it assumes that the problem with the watermark is because *you* are an unauthorized person trying to tamper with the file! It will refuse to open the file for you. This is your warning that the file has been tampered with. OpenGrade will also let you view a report that shows more information about the tampering, e.g., which lines were tampered with, deleted, or added.

In everyday use, you should go by the following rules to protect your files. Make frequent backups and printouts. Don't store your gradebook files on a computer that people can get access to without logging in. Never leave a file open in OpenGrade if you're getting up from your desk. Use a good password: one that is at least eight characters long, isn't a dictionary word, and contains both letters and digits. Don't write your password down, and don't use passwords people can guess, such as your license plate number, your kid's birthday, or your pet's name. Don't type your password if someone can look at your hands and see your keystrokes. Don't give anyone access to copies of your gradebook file, be-

cause a determined hacker might be able to determine your password by writing a program to guess millions of possibilities.

Here are some real-life situations that can arise, and methods for dealing with them:

You try to open the file, and OpenGrade doesn't accept your password. Don't panic! The most likely cause of this is that you've made a mistake typing in your password. Just try again.

You've forgotten your password, and you need to edit the file. OpenGrade does not store your password anywhere, so there is no way to get it back. Use Change Password under the File menu to change the file's password.

You need to look at an old OpenGrade file from years ago, and can't remember the password The easiest solution is to examine the file in a text editor. Otherwise, change the password as described above.

A student tampers with your file using a text editor. Once you've convinced yourself that you didn't just make a mistake typing in your password, here's what you should do. Click on the button to view the report on the tampering, and save the report to a file. Next, make a copy of the file. You'll probably also want to print out both of these files. Next, restore your gradebook file from your latest backup. (If you've been uploading your grade reports to a server, then you should have a recent backup copy there.) By examining the report on the tampering, you should be able to repair some or all of the damage, and you may be able to figure out who the tamperer might be, e.g., if all the changes were increases in a particular student's scores.

A student changes the password of your file. This could happen if the student has read this documentation! You'll find out that the file has been tampered with, because OpenGrade will refuse to open it with your password. The procedures for handling this are similar to the ones described in the preceding paragraph. However, you won't be able to get a line-by-line report on the tampering.

This is too much security for you. You don't want to be bothered with passwords. Just use a password with no characters in it. All you have to do is click on the OK button whenever OpenGrade asks for a password.

This isn't enough security for you. Download Gnu Privacy Guard (also known as GPG or GnuPG), and use it to encrypt your gradebook files.

8 Posting Grades on the Web

This section describes OpenGrade's machinery for posting grades on the web.

To post grades on the web, you'll need access to a server where you can install files and CGI software yourself. Install the `ServerOG.cgi` script distributed with OpenGrade. To view their grades, students will interact with a second CGI script, `Spotter.cgi`, which is distributed with the Spotter software, also on the `lightandmatter.com` site. You will need to set up a directory tree on your server as described in the Spotter documentation, in the section titled "Interfacing."

Once this is all set up, you can upload grades by selecting Post Grades from the Server menu.

9 How Grades Are Calculated

There may be situations where it's not obvious to you how the software is calculating students' grades. This may happen when you have categories in which you drop some low scores, or when you have assignments that aren't due yet, but which some students have already turned in. OpenGrade has definite mathematical rules, documented in this section, for handling such situations. Although the rules are a little complex, there is one general principle that will usually keep things clear in your head: the number of possible points is always the same for every student. This principle has two advantages: (1) If we want to know who is doing better in the course, student A or student B, we don't have to specify whether we mean better in the sense of total points or better in percentage terms. (2) It makes sense to compute statistics like averages and standard deviations for the class as a whole, and these statistics can be computed in terms of points, not just in terms of percentages.

dropping low scores When dropping low scores, we choose them so as to maximize the student's total points, not the student's percentage. The maximum number of possible points in the category is calculated by leaving out the assignments that have the lowest number of points possible. Thus, the numbers deleted from the list of scores do not necessarily correspond to the numbers deleted from the list of possible points. If there is extra credit on some assignments (i.e. the student has scores higher than the maximum), then these extra credit points are skimmed off the top before any of the calculations are carried out, and then added back on at the end.

due dates When you create a new assignment, the default is to leave the due date blank; the software then assumes that the assignment has already fallen due.

The only time you really need to enter a due date for an assignment is when it's something that some students will turn in early. You only need to enter the month and day, not the year; the year is figured out based on the date you gave for the beginning of the term when you set up the file. If an assignment isn't due yet, then it doesn't count in students' grades at all. This is in keeping with the general principle that the number of points possible is the same for all students. Thus, a student may turn in an important assignment early, know she had a certain grade on it, and know that her grade in the course is going to change a lot on the assignment's due date; but the effect will not show up in the software until the due date.

An assignment is considered to have fallen due at the beginning of the day it was due. For instance, if June 3 is the day you'll be turning in final grades for your course, then you can set the due date on a term paper to June 3, and it will be included in the grades you calculate on June 3.

weighted grading You can set up a gradebook file for either weighted or point-total grading; this is determined when you make your first category.

With weighted grading, if the total number of points possible in a particular category is zero, then that category is treated as if it doesn't exist, and the other categories then become more important. Because of this, there can be no such thing as an extra credit category when you use weighted grading.

10 Format of Gradebook Files

OpenGrade's files are in a plain text format that you can edit in a text editor. The following is a nearly minimal example.

```
1  class
2    .title "B. Crowell's Physics 205/210, spring 2002"
3    .staff "bcrowell"
4    .days "MW"
5    .term "2002-1"
6    .dir   "bcrowell/s2002/205"
7    .standards "A:90","B:80","C:70","D:60","F:0"
8
9  preferences
10   .backups_on_server "true"
11
12 preferences
13   .file "/home/bcrowell/Library/Preferences/OpenGrade_prefs"
14
15 categories
16   .e "catname:exam,exams","max:120"
17   .hw "catname:homework,homework","drop:1"
18
19 roster
20   .newton_ike "id:1234"
21   .einstein_al "id:5678","dropped:true"
22   .oflaherty_karen "id:3333","last:O'Flaherty"
23
24 assignments
25
26 grades
27
```

Lines 2-5 give general information about the class. The title is placed at the top of web reports. The staff list gives usernames of the instructors and TA's who may make changes to the gradebook. The list of days the class meets is used to make it easier to enter certain dates, and likewise for the line specifying the date on which the current term began.

Line 6 gives a subdirectory of the cgi-bin in which this class's grades will be stored. Line 7 sets your grading standards for this class. On reports, these symbols will be displayed along with the numerical scores. If you don't want any grade symbols displayed, you can just leave the rest of the line blank after .preferences. The symbols are entirely arbitrary.

Lines 9-10 tell where your preferences file is located. OpenGrade tries to put this in the customary location for your operating system.

Lines 15-17 specify that there are two categories of graded work in this class: exams and homework. The short tags e and hw are used internally, and are also used in the keyboard-based terminal interface when the user needs to select a category. The catname parameter give two forms of the name, which are usually singular and plural. The first one is used in constructing strings like exam 1 for use in reports, and the second one is for headings in reports. In line 16, we specify that exams are worth a maximum of 120 points by default;

this is optional. In line 17, the parameter `drop:1` says that we drop the lowest homework score.

Lines 19-22 give the roster. Normally the names are given simply as database keys, in the format `lastname_firstname`. When the name is displayed, the first character of each name is capitalized. When this is not the correct capitalization, or when the name contains punctuation, the correct form can be indicated separately, as in line 22. Line 21 indicates that Al Einstein has been dropped from the course, so he will not show up on reports. However, his data are still in the file, and he can be added back in at any time.

Lines 24 and 26 are placeholders for the assignments and grades. These can be added later.

This listing is different from the OpenGrade files you'd typically see, because it lacks authentication codes. Authentication adds some data to the file which is preceded by a `#` sign wherever it occurs. Every line has some authentication information at the end, and there are some separate authentication lines at the beginning and end of the file. The format of the authentication codes is described in the source code of the `LineByLine` module.

11 Future Improvements

general There's a CPAN module for exporting to Excel format. I should implement this, and also ones for GNUMERIC and Kspread. Make `File::Import` and `File::Export`. Support tab-delimited and comma-delimited files?

It should be possible to get stats like what percentage of the class has A's, etc. This could be done by adding more outputs to the `Crunch::stats` subroutine.

`MyWords.pm` has a mechanism for defaulting to English if the translation in the preferred language isn't available. This should be generalized so that the fallback isn't always English, and it should be in `Words`, not `MyWords`.

Some server operations should be made more efficient. When uploading grades, compression could be used. When listing work from the server, the algorithm that runs on the server shouldn't read each file more than once.

GUI Some of the more important properties of the currently selected category, student, and assignment should be displayed in the main window. There should be a way to get the other information as well.

In the dialog box for adding a new category, there's some extra help that the `TermUI` gives when weights are enabled, and the GUI should give that help, too. Should also fill in plural based on singular, if they haven't typed anything in plural yet. The box should be taller so you don't need to use the scrollbar.

When creating a new file, there needs to be a way to set whether weighted grading will be used. (`TermUI` does this when you add the first category, but don't do it that way in the GUI, too awkward. The decision really needs to be made when they open the file.) There is no warning when you create a new file on top of an old one.

In `FileDialog`: (1) Inputs should look sunken, not raised. (2) When creating a new file, the focus should be in the right place to start with. (3) There's no easy way to set `activeBackground`; in KDE, the default is lame, and this causes buttons to become totally black when the cursor is over them.

Missing functionality: change order of assignments or categories, change preferences.

Under the Assignments menu, there should be a way to change the weight of a category.

The assignments listbox should start out positioned at the end, not the beginning, when you select a new category. This seems to be pretty difficult to do, however! Selecting and then unselecting the last item doesn't work, and it doesn't have `set()` and `get()` methods like a Scrollbar widget.

The dialog box for creating a new file doesn't give all the help that the text-based version does. It doesn't check whether the days of the week are legal. Actually, I think I can do this by building new types and options in to the Input class.

Should allow the user to view dropped students. Have a note in `add_or_drop()` about this.

12 Bugs

general The concept of dropping the n lowest assignments doesn't make sense when the assignments are of unequal value. This can have a particularly goofy effect when you have a whole bunch of assignments that are all worth the same number of points, and only one assignment that's worth a very small number of points; the total possible is computed by throwing out the one that's worth the fewest points, which probably isn't what was intended. This is the way it has to be based on the principle that the number of possible points is the same for all students, but it's not intuitively obvious. There should be a warning message or something, even though it's just doing what the documentation says it does.

Names of assignments work fine when they're numeric, but not when they're real names. Need a mechanism to construct both a database key and a separate name for reports. Right now, I just convert non-word characters to underbars. Also, there are a zillion places in the code right now where I do something like `"$category $assignment"` to make a human-readable name for an assignment; this should be a method in GradeBook.

Some errors should be checked for when reading in the gradebook file: two students with the same key; mangled syntax in parameter lists; grades for a student who isn't on the roster.

When you create a new category, it should require you to give a weight if you're already using weights, or prohibit you from doing so if you aren't already. This should really be selected when you create the file.

With some files, I couldn't do any client-server stuff until I touched the client-server settings. It didn't matter that I wasn't even changing the settings.

Downloading grades: Some hardcoded English text. Needs more flexibility for scoring.

All client-server transactions should be wrapped in a subroutine. The subroutine should check whether the password works, and if not, prompt for the right one. If not connected, should give an error message. It should then reconcile the server and client: server's e-mail overrides client's, and all other data from client overrides server's: disabled accounts,...? Should also handle errors. Need to deal with cases where the keys for the same student on the server and client become inconsistent.

The time zone correction in `ServerOG.cgi` when downloading grades may be incorrect if the client and server are in different countries that have different definitions of daylight savings time. It will also not work correctly if either time zone is a non-integer.

When posting grades and uploading the backup copy of the gradebook file, the version of the gradebook uploaded is the one on disk, not the current one in memory.

There is now the Preferences module, but there's also a preferences mechanism in the GradeBook module. Should reconcile these two. Everything should be orthogonal: every pref can be stored in either the global prefs file or in the gb file, and the one in the gb file overrides the global one. All prefs currently default to false, which is wrong, e.g., for `backups_on_server`.

Test whether the Preferences file works correctly in Windows.

When downloading work, due date should be easier to edit, and it should be harder to forget to set it correctly. The algorithm is a kludge: AND together the "find" parts.

It's possible for a due date for an assignment to be written to the file without any year on the front, and OpenGrade then chokes and crashes when you try to open the file again. I think this might happen when the due date is in the same month as the beginning of the term.

GUI When you type in a score with a trailing x to defeat the warning about extra credit, the x should automatically get edited out after the focus goes somewhere else.

A bunch of the dialog boxes that you pop up from the server and reports menu have blank or bogus titles.

`ExtraGUI::show_text()` should warn you when you try to save its results on top of a preexisting file.

The GUI allows you to pick assignment names like "1&2," containing the illegal character &.

When you select a student using the keyboard, the roster should scroll if necessary.

There is some GUI stuff relating to passwords in `BrowserData::new()`.

On Windows, if you click on an assignment, the assignment isn't selected in the right-hand half of the window; instead, overall grades are displayed. The colors in the roster also look strange on Windows.

When you first open a file, some inefficient stuff happens: first it builds the roster one way, then another.

The dialog box for setting standards is too tall when you use ABCDF+- grading.

I wrote stuff into the `Browser.pm` package at the outside level, assuming there would only ever be one window open. This needs to go into `BrowserWindow`. Likewise, what happens with stuff like `grades_queue()`, which is inside a `BEGIN{ }` block? Will its variable get properly initialized when I make a new `BrowserWindow` object? Will each `BrowserWindow` object really have its own `%queue`?

`GradeBook::add_assignment()` has some hardcoded English text, and there is also some in `Report::web()`.

When opening files, the buttons aren't properly internationalized. This can be done using some of the options to `Tk::FileDialog`, but I just haven't done it yet.

Input class needs 'date' and 'yn' types. Use yn type in dialog box for creating a new category. See `edit_assignment()` in `Browser.pm`.

A bunch of the menu items should have "..."

Resizing needs to be handled better. Some widgets have hardcoded sizes, which I'd like to avoid.

text interface The raw string "weight" occurs in `TermUI.pm`.

The terminal-based interface should figure out the width of your terminal window, rather than defaulting to 80 columns.

Emacs-style tab completion is goofy in a case where one student's name is the same as the beginning of another's (Ng and Nguyen) — you need to type the underbar!

In the statistics report, there is no user interface to say if you want stats within a category, and the report routine doesn't support that, either. The formatting of the report could be improved a lot.

When selecting a student, should be able to type in first name. The display is also kind of messed up during the process of selecting a student. And the underbar should be hidden.

Give some message when they make a choice that's not on the menu, or not allowed because there is no open file, etc. In main loop, can make hashes to control the behavior of the commands.

Menus should allow you to type multiple characters in a row and then hit return so you go into a sub-menu. (Don't just take keystrokes without hitting return, because some commands like "revert" are more than one character.) Beep when they do an illegal command, and flush the rest of the string out.

13 Programming Stuff

version history 0.1 — Initial release. There is only a text-based user interface. March 2002

0.2 — The text-based interface is pretty solid. There is a graphical user interface, which has limited functionality and hasn't been tested much. April 2002

0.3 — The GUI has reasonably complete functionality, but still needs testing. April 2002

0.4 — The GUI is fairly solid. April 2002

0.5 — Improvements to the GUI. Bug fixes. April 2002

0.6 — Password authentication. Improvements to the GUI. Bug fixes. May 2002

internals You can use `make internals` to make the file `internals.html` from the POD documentation.

14 Licenses

Copyright (c) 2002 Benjamin Crowell, all rights reserved.

This software is available under version 2 of the GPL license. The software is copyrighted, and you must agree to the license in order to have permission to copy it. The full text of the license is given below.

GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice

placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with

the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRIT-

ING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the program's name and a brief idea of what it does.
Copyright
(C) year name of author
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This is free software, and you are welcome to redistribute it under certain conditions;
type 'show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate

parts of the General Public License. Of course, the commands you use may be called something other than ‘show w’ and ‘show c’; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program ‘Gnomovision’ (which makes passes at compilers) written by James Hacker.

;signature of Ty Coon, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

The GNU Free Documentation License

Version 1.1, March 2000

Copyright © 2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other written document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft,” which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. Applicability and Definitions

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you.”

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and

contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not “Transparent” is called “Opaque.”

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, \LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

2. Verbatim Copying

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. Copying in Quantity

If you publish printed copies of the Document numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts

on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. Modifications

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- State on the Title page the name of the publisher of the Modified Version, as the publisher.
- Preserve all the copyright notices of the Document.
- Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

- Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document’s license notice.
- Include an unaltered copy of this License.
- Preserve the section entitled “History,” and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- In any section entitled “Acknowledgements” or “Dedications,” preserve the section’s title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- Delete any section entitled “Endorsements.” Such a section may not be included in the Modified Version.
- Do not retitle any existing section as “Endorsements” or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section entitled “Endorsements,” provided it contains nothing but endorsements of your Modified Version by various parties – for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. Combining Documents

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled “History” in the various original documents, forming one section entitled “History;” likewise combine any sections entitled “Acknowledgements,” and any sections entitled “Dedications.” You must delete all sections entitled “Endorsements.”

6. Collections of Documents

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. Aggregation With Independent Works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an “aggregate,” and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document’s Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. Translation

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. Termination

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. Future Revisions of This License

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License."

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.