1 Introduction

This document is intended for CSE teaching staff. As an instructor or TA, you must assume certain responsibilities for setting up and maintaining the instructional resources for your courses. Those responsibilities are outlined here. A companion document, Preparing CSE Instructional Resources for Technical Staff, outlines the responsibilities of technical staff.

Guide for New Instructors is a useful resource for getting new instructors up to speed. Contact the course coordinator or webmaster to get access to that for appropriate users.

The resources discussed here include:

- Instructional hardware and software computing facilities.
- Groups and accounts for instructional staff and students.
- Disk space for holding course directories. There are three types of directories, designed to hold
  1. “general purpose” course files,
  2. course web documents on the CSE WWW server, and
  3. student projects.
- Archives of course directories, available either on line or on CDs.
- Course mailing lists.

If you see text like this in the following sections, take note: it points out one of your responsibilities in maintaining instructional resources.

2 Hardware and Software Facilities

2.1 Instructional Labs and Servers

The primary purpose of the instructional labs and servers is to support the instructional goals of the department. See http://www.cs.washington.edu/lab/facilities/instr-labs.html and its links for descriptions of the hardware and installed software of all workstations and servers dedicated to instructional support.

Note that the computing environment on both Linux and Windows instructional workstations and servers is not exactly the same as the environment on research workstations and servers. The instructional environment, for example, includes some tools that are used only for classes and are not normally available on research hosts, while the research environment allows access to a much wider range of /projects/... directories.

Hence, except for web-based tools, you should always log into an instructional host when running the setup procedures described in this document.
2.2 Instructional Workstation Testbed via Remote Desktop

In room CSE581 is a workstation, `thage.cs.washington.edu`, configured identically to the Windows XP workstations in the instructional labs. Thage can be used by teaching staff who need to test instructional software, but who would prefer to do it in their office via remote desktop.

2.3 Quarterly Course Computing Requests - Response Required

Instructors for each course need to inform CS Lab Staff of special computing needs before the start of each quarter by sending requests to `course-computing@cs`. Usually there is a reminder email message sent to instructors well before each quarter begins.

You must respond to this reminder, even if it is to say you need no special software or hardware for your course.

You can see course requirements for the current and previous quarters by following the links of [http://www.cs.washington.edu/lab/course-computing](http://www.cs.washington.edu/lab/course-computing).

2.4 Free Microsoft Software for Students

All CSE majors and any student registered in a CSE course may find it convenient to load MS software, in particular development software such as Visual Studio, on their personal computers. See—and point your students to—[https://www.cs.washington.edu/lab/software/](https://www.cs.washington.edu/lab/software/) for complete information.

3 Groups and Accounts for Instructors, TAs, and Students

3.1 Groups

Each course has at least one associated administrative group, e.g. for CSE000 the group name would be `cse000`. In most cases such course groups are automatically initialized before the start of each quarter, based on a list of courses and instructors. Initialization consists of the following:

- The group is created if necessary.
- The list of “owners” is adjusted if necessary to include the instructors for the upcoming quarter.

Instructors must use the Group Admin tool to add other owners and members to the main course group. Also use this tool to delete owners and members that no longer need administrative access to your course.

In rare cases the group may not yet have been created; in this case please contact support@cs to get it created and initialized with the proper owner(s).
Other groups can also be associated with a course; these include:

**student project groups**

Usually called `cse000l`, where `l` is a letter, a to z. Project groups are created with the main course group as their “master” group\(^1\), so that ownership of project groups will be determined by ownership of the main course group.

**course administrative groups**

Rarely, you might want to use group permissions to restrict access of some directories to a subset of a course’s teaching team using a special group, e.g. `cse000-admin`.

Notify support@cs if you need additional groups created.

An overall group, `tas`, is also established at the start of each quarter. It consists of TAs who are registered one week before the start of a quarter on the TA assignments page, [http://www.cs.washington.edu/education/ta-home/ta-assignment](http://www.cs.washington.edu/education/ta-home/ta-assignment). One of the purposes of this group is to allow TAs to carry out duties not normally permitted to undergrads, such as reserving equipment and rooms.

Note that all groups are “universal” with respect to the systems we run: the same groups exist—with the same members—on all Unix and Windows instructional systems.

### 3.2 Accounts

Except for new visiting lecturers, instructors and CSE majors should already have computer accounts on all general purpose instructional systems. If you are supervising visiting lecturers who do not yet have accounts, direct them to the account application form and tell them to fax you the filled out form.

There is a guide for new instructors MediaWiki with information that is specifically tailored for new instructors. That wiki uses built-in application auth specifically so that CSE accounts are not a prerequisite for users to access it. New instructors should have been contacted by the course coordinator (currently pl at cs) with their credentials, but webmaster can also create accounts there.

This section deals mainly with special purpose accounts and systems; these are listed on the Course Computing page for the current quarter.

Instructors can, in addition, request permanent accounts for themselves on special purpose instructional systems.

In order to protect themselves against possible malicious code, those TAs who grade programming assignments may also request an additional temporary instructional “sandbox” account for themselves by contacting support@cs.

Non-majors who take CSE courses will in most cases be given temporary accounts automatically, based on course enrollment.

\(^1\) see [http://www.cs.washington.edu/lab/GrpAdmin/grpann.html](http://www.cs.washington.edu/lab/GrpAdmin/grpann.html) for an explanation of the concept of master group
Except for faculty and staff, special purpose instructional accounts are temporary: they are terminated either the day after the last day of a quarter, or, for TA accounts, a week or two after a quarter ends.

You should inform non-majors that they can pick up their account information at the front desk.

Important note: Non-majors who register for a course after the first day of classes will not automatically get accounts. In this case the teaching staff must send a list of the late registrants to support@cs. This list will then be forwarded to the undergrad advisors.

The following instructional systems require temporary accounts to be set up.

**linux systems for OS and networking courses, e.g. cse451**

These have been variously named *forkbomb*, *coredump*, *spinlock*, and *blackboxnn*. They are configured as directed by the teaching staff.

**web and DB servers**

The Linux web and DB server is *cubist*.

Special local accounts are set up for members of each class needing access.

**vole, the NFS gateway for undergrad and non-major TAs**

See Section A.1.1 [Course Web Access for Undergrad and Non-Major TAs], page 12. TAs registered on the *ta-assignment* web page at least one week before the start of a quarter will automatically be given temporary accounts on vole. Such accounts will be expired two weeks after the end of the quarter or just before the start of the following quarter, whichever comes first. Any TA subsequently added to the teaching staff can get a vole account by simply filling out an *account application form*.

### 4 Course Files

#### 4.1 Naming Conventions

In the following sections we will assume the course you are dealing with is CSE000. Several directory names will be based on this course name. Additionally, directory names will be based on quarters, using the convention *yyqq*, where *yy* is the year (**05**, **06**, etc.), and *qq* is the quarter (**au**, **wi**, **sp** or **su**). For example, */cse/www/education/courses/cse000/13wi* is the location of CSE000’s web for the Winter ’06 quarter.

When directories and files are mentioned below, only Unix path names are given. The equivalent Windows paths are shown in the following table:
<table>
<thead>
<tr>
<th>Unix_path_prefix</th>
<th>equivalent_Windows_path_prefix</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>/cse/courses</td>
<td>\cseexec\cs\cse\courses</td>
<td></td>
</tr>
<tr>
<td>/projects/instr</td>
<td>\cseexec\cs\unix\projects\instr</td>
<td></td>
</tr>
<tr>
<td>/cse/www</td>
<td>\cseexec\cs\cse\www</td>
<td></td>
</tr>
<tr>
<td>/cse/www</td>
<td>\cseexec\cs\unix\vole\cse\www</td>
<td></td>
</tr>
</tbody>
</table>

Thus, /cse/courses/cse000/13wi in Unix is equivalent to \cseexec\cs\cse\courses\cse000\13wi in Windows. If you use Windows you may find it convenient to map drive O: to \cseexec\cs; see [Course Web Access for Undergrad and Non-Major TAs], page 12.

### 4.2 General Purpose Files

Course files for CSE000 are found under directory /cse/courses/cse000. The following subdirectories always appear under this directory:

#### yyyy

This is the notation described above, e.g. /cse/courses/cse000/13wi. These directories hold files or software packages that change from quarter to quarter. What is stored here is at the discretion of the CSE000 instructional staff—anything except student projects.

Note that by default this directory is readable by all, and writable by group cse000. If you want to put files here that you don’t want your students to see, you should create a directory under yyyy that has world read permissions removed, then put those files under that protected directory.

#### turnin

This directory is reserved for the electronic assignment turnin program. See the project(1) and turnin(5) man pages on one of the instructional Unix servers for usage instructions.

Only projects already in the database can receive submissions sent by students using turnin, as well as further modifications using project, e.g. enabling and disabling submissions.

In order for this to work, an instructor must initialize a project by running `project -i`. This will cause a request to be sent to support@es; support personnel will then add the project to the projects database. If you override the default turnin directory, be sure to specify a full path name, i.e. one that begins with ‘/’.
Preparing CSE Instructional Resources for Teaching Staff

/cse/courses/cse000 and its immediate subdirectories are normally readable by all and writable by group cse000.

You can ignore /cse/courses if you have no need for such storage and you do not plan to have your students use turnin(1).

4.3 Web Documents

4.3.1 Course Home Pages

Every course needs to have a “home page.” Links to all these pages are found at http://www.cs.washington.edu/education/courses/.

A schematic of the course web file structure for CSE000 looks like this (assuming the current quarter is Winter '13):

```
/cse/web/courses/cse000/index.html
  " /13wi/index.html
  " " /additional web files for 13wi
  " /yyqq/index.html
  " " /additional files for qtr yyqq
```

In the above, yyqq represents course webs for all quarters previous to 13wi. A similar directory hierarchy exists for each course.

Note that there is a “per-course” web page, /cse/web/courses/cse000/index.html, whose advertised URL is http://courses.cs.washington.edu/courses/cse000/. This per-course page is automatically renewed for CSE000 each quarter. You can see a typical example of a per-course web page at http://courses.cs.washington.edu/courses/cse457. By default it contains a description of the course (taken from the official course catalog), links to the current quarter and previous quarter webs, and links to the home pages of prerequisite courses. The per-course page for CSE000 can be edited by members of group cse000, but only within strict constraints; see Section A.1.2 [Editing Per-Course Webs], page 12 in the Appendix for more information.

For brand new courses, contact support@cs to request the creation of the course web.

Email addresses should not be included in any web content. See http://www.cs.washington.edu/lab/services/spam/leakage.html for info on how to display such addresses without “including” them.

See http://www.cs.washington.edu/lab/auth.html if you need to restrict access to your course web; by default a course web tree is visible to the world.

---

2 The web server is configured to look for index.html if given just a directory name. In web server jargon index.html is called a “default document.”
Before each academic quarter starts, a “per-quarter” directory for CSE000 will be automatically created, e.g. /cse/web/courses/cse000/13wi. The corresponding URL is http://courses.cs.washington.edu/courses/cse000/13wi.

You must create a “default document” page, index.html or index.htm, in the current quarter course web directory. If it is not created, users will get “document not found” errors when they try the published URL. The design of this page and all the other pages in the per-quarter directory is entirely at your discretion—there are no standards imposed for how per-quarter course webs should look.

Note that for convenience a shorter URL, http://courses.cs.washington.edu/cse000, will be redirected to http://courses.cs.washington.edu/courses/cse000/13wi/.

/cse/web/courses/cse000/ is owned by some common instructor of the course, has group ownership set to cse000, and is group writable. Each per-quarter web directory is owned by the instructor of CSE000 for that quarter.

In order to allow collaboration of the cse000 teaching team in maintaining the course web, all files that are writable by owner in the course web should also be writable by the associated group: a cron job runs nightly to enforce this. Members of cse000 who work from Windows should not have to worry, as the permissions of a newly created file or directory are inherited from the containing directory.

However, instructional staff who work from Unix should ensure group writability of newly created files and directories by using a Unix “umask” of 002 (set by shell builtin command ‘umask’—see tcs(1)).

See Section A.1.3 [Support for CSE590 Webs], page 13 if you need to create a 590 web.

See http://www.cs.washington.edu/lab/auth.html for information on controlling access to course web resources.

4.3.2 Course WIKIs

Some courses have set up WIKIs to publish course material and, perhaps, to allow feedback from students. See the instructional course URL "Main Page" for a list of course WIKIs.

Some information about MediaWiki, the software used for lab-supported CSE WIKIs, can be found at http://www.cs.washington.edu/lab/www/MediaWiki.shtml. That document explains what policy decisions need to be made before a new install can be created.

If you want to set up a WIKI for a course that does not already have one, contact support@cs.
4.4 Student Project Areas and Repositories for Version Control Systems

A special shared directory under /projects/instr/yyqq (where yyqq denotes the quarter as described in Section 4.1 [Naming Conventions], page 4) is created every quarter for course-related student projects. This space is not subject to the quotas imposed on student home directories. It is most often used by teams of students working on group projects. For CSE000, the project directory would be /projects/instr/yyqq/cse000, e.g. /projects/instr/13wi/cse000.

A directory link /projects/instr/CurrentQtr is also maintained, where the meaning of “CurrentQtr” is somewhat fluid.³

A project directory for a given course and quarter is created only upon request of the instructor or TA.

Note that student projects, which can consume quite a bit of disk space, should not be stored under /cse/courses/cse000/13wi. If projects were put there the number of academic years of /cse/courses than can remain in the on-line archive would be decreased.

If you want to establish a project area for your course, please let support@cs know the course name and the number of project teams. The top level project directory will be created for you, along with the necessary group names, if necessary (e.g. cse000a, cse000b, etc.). It is YOUR (instructor or TA) responsibility to create subdirectories for each project team. See Section A.2 [Creating and Setting Ownership and Permission of Project Directories], page 14 for more information and suggestions on how to get this done.

Older project files are kept for about a year.

Please notify support@cs if you want to keep your courses’s project files on-line for longer than a year.

When student project areas are requested, teaching staff also frequently have questions about setting up repositories for versioning systems like Subversion or CVS, in particular questions about accessing repositories from non-CSE machines. For CSE machines the authentication method is just file access, e.g.

- For Subversion, a repository URL would be file:///projects/instr/... under Linux or file:///cseexec/cs/unix/projects/instr/... under Windows.
- For CVS, the CVSROOT environment variable is just set to a path name: /projects/instr/... on a Linux host or \cseexec\cs\unix\projects\instr\... on a Windows host.

For machines outside the department, ssh is the only authentication method supported:

- For Subversion, a URL would be svn+ssh://attu.cs.washington.edu/projects/instr/....

---

³ The current quarter means variously “being taught right now,” “most recently taught,” or “about to be taught,” depending upon when the link gets updated.
• For CVS, use “ssh” as the authentication method and set CVSROOT to
  \textit{username}@\textit{attu.cs.washington.edu}/projects/instr/...\textsuperscript{4}

4.5 Archives

4.5.1 General Purpose Archives
Quarterly course directories for previous academic years, e.g. /\textit{cse}/courses/cse000/01au, are moved to archival disk storage; however, through the magic of symbolic links the path names do not change. As on-line archival disks fill up, the oldest academic year archive is written to CDs and then deleted. Generally such archives are kept on-line for several years.

4.5.2 Course Web Archives
Course webs older than one academic year are moved to archival disk storage. Like general purpose files, through the magic of symbolic links neither paths nor URLs change when webs are archived.

4.5.3 Student Project Archives
Projects are not routinely archived.

\begin{quote}
If you need specific student projects archived to CD notify support@cs.
\end{quote}

4.5.4 Archive Permissions
Archives are nominally read-only. However, write-access by tech staff is possible because instructional staff sometimes forget to read-protect or remove certain files from the archives, e.g. answers to homework assignments

\begin{quote}
• Remember to make changes to course file contents and permissions before archiving takes place.
• Contact support@cs if you find that you do need to make changes to content in archival storage and specify exactly what you want changed and how to change it.
\end{quote}

\textsuperscript{4} For example: If your students are coming from another linux box, tell them to set up these environment variables:

\begin{verbatim}
export CVSROOT=\textit{username}@\textit{machine}:\textit{repositorypath}
export CVS_RSH=ssh
\end{verbatim}
then issue their CVS commands as usual.

If they are using WinCVS on a Windows box, tell them to use pointy-clicky to navigate as follows: Admin -> Preferences... -> General. Choose “ssh” as the authentication method, then enter

\begin{verbatim}
\textit{username}@\textit{machine}:\textit{repositorypath}
\end{verbatim}

in the CVSROOT text box

where \textit{machine} is usually \textit{attu.cs.washington.edu} (collective name for the instructional linux servers) and \textit{repositorypath} is /\textit{projects/instr}/...
5 Course Email Lists

When setting up a course mailing list, the instructor has two main options:

1. Go to MyUW, click on “Request class email list”, and get an @u.washington.edu mailing list. This list is populated with the students (and instructors of record), and is automatically updated as students add and drop the class.

2. Manually administer a mailing list at @cs.washington.edu. NOTE: Use of the CSE mail lists described in this section is deprecated. Instead, please use the UW maintained mailing list, whenever possible.

Most instructors choose option 1, which is more convenient. This document describes option 2. Option 2 is mainly a throwback to the days before LST (Catalyst tools) provided course mailing lists, when creating a mailing list at @cs.washington.edu was the only choice.

Help with setting up the UW mail list is also mentioned on each instructor dashboard page, e.g., http://courses.cs.washington.edu/courses/cse000/13wi/admin/.

As in the previous section, we assume your course is CSE000. The associated mailing list, cse000@cs.washington.edu, as well as all class lists, are created and maintained by Mailman, the free web based mail management tool from GNU.

Those who have a valid CSENetID and are members of group cse000 can manage cse000@cs via the list administration page, http://mailman.cs.washington.edu/mailman/admin.

Another useful Mailman link is https://mailman.cs.washington.edu/mailman/listinfo?ads=cse, which shows all CSE mailing lists.

Note that management of a class mailing list includes the ability not only to administer the list, but also to create and delete it. When you delete a list for which an archive of messages is kept, Mailman will ask whether you want to delete the archive as well; if you choose not to delete the archive, that archive will be re-associated with the list when it is created anew.

You need to follow the steps below in order to maintain the mailing list for CSE000:

1. If cse000@cs already exists, remove it. To do this visit the mailman administration page, click on cse000, then either
   - Click on Delete this mailing list and confirm it. Note: due to a glitch in Mailman you may need to make yourself the administrator (owner) of the list in order to delete it; otherwise it may give you a “not authorized to delete” error.

---

5 Instead of a class mailing list hosted at cs.washington.edu, you may choose instead to create one hosted at u.washington.edu. From MyUW follow the “Request class email list” link; these class lists are briefly described in http://www.washington.edu/computing/mailman/faqs/classlist.newowner.html. Most instructors choose the cs.washington.edu option.
- Or, click on **Membership Management**→**Mass Removal** and remove the current members.\(^6\)

2. If necessary, (re)create the class mailing list for CSE000 by visiting [http://mailman.cs.washington.edu/mailman/create](http://mailman.cs.washington.edu/mailman/create).

3. Update the “list administrator email addresses” in the **General Options** page with the correct list for the current quarter.

4. If you want to create a **cse000** email archive, go to its administration page, click **Archiving Options**, and choose the appropriate options.

5. To add people to **cse000@cs** go to its list administration page. Click on **Membership Management**→**Mass Subscription**, then add email addresses to the box there or upload them from a file. The addresses must be fully qualified, e.g. `jouser@cs.washington.edu`, and listed one per line. You can download a list of all class registrants using the CSE Enrollment Tool.

Alternatively, you can delegate management to the students in CSE000. By default, a class mailing list allows people to add themselves after it is created.

6. Finally, review **Privacy options**, which set up policies on who can subscribe to your class list (**Subscription rules**), who can post messages to the list (**Sender filters**), and what kind of mail should be passed through, moderated or discarded (**Spam filters**).

7. You can create additional mailing lists for your course by following the above steps. For CSE000 you can create any list whose name starts with ‘**cse000-**’, e.g. **cse000-staff**.

You will need to know about the following URLs for **cse000**:

**list “home page”**

Students can visit the list home page, [http://mailman.cs.washington.edu/mailman/listinfo/cse000](http://mailman.cs.washington.edu/mailman/listinfo/cse000) to add themselves, or they can use the email interface to subscribe. Email to **cse000-request@cs** with the word “help” in the subject will return a message listing all of the email command options.

**list archive page**

Students can also visit the list archives by clicking on the very first URL on the list “home page.” Note: For public archives, this URL would be [http://mailman.cs.washington.edu/pipermail/cse000/](http://mailman.cs.washington.edu/pipermail/cse000/); for private archives, viewable by list members only, it would be [http://mailman.cs.washington.edu/mailman/private/cse000/](http://mailman.cs.washington.edu/mailman/private/cse000/).

---

\(^6\) At present **Mailman** does not list the current members in the deletion window, so this way of deleting the members is practically worthless. Until this gets fixed, the best way is just to delete and re-create the entire mailing list.

---

**Appendix A** Appendix
A.1 Specialized Course Web Access

A.1.1 Course Web Access for Undergrad and Non-Major TAs
A Unix server, \texttt{vole}, has been provided as a gateway for editing files under \texttt{/cse/www}. \texttt{Vole} differs from the standard undergrad Linux servers (e.g. \texttt{attu}) in the following two ways:

1. Only TAs, and technical staff have login accounts there.
2. The web server exports (via NFS) \texttt{/cse/www} to \texttt{vole}.

So, if you are an undergrad or non-major TA, you can access \texttt{/cse/www/education/courses/cse000/13wi/index.html} and siblings in one of two ways:

1. Log into \texttt{vole} (just as you would log into \texttt{attu}) and you will have access.
2. From your instructional (CSEPCLAB domain) Windows account you can access \texttt{\csexec\cs\unix\vole\cse\www}, as described in \url{http://www.cs.washington.edu/lab/sw/uwcsentdfs.html#gateways}.

You may find it convenient to map your course’s per-quarter web, \texttt{\csexec\cs\unix\vole\cse\www\education\courses\cse000\13wi} to a drive letter.

\begin{center}
TAs should use only the web editing tools that have been approved by their supervising faculty.
\end{center}

A.1.2 Editing Per-Course Webs
This section considers course “home pages,” or per-course webs, an exemplar of which can be seen at \url{http://courses.cs.washington.edu/courses/cse457}.

If you are satisfied with the default appearance of your per-course web, you do not need to do anything and you can skip this section.

However, you \textit{can} edit it within certain constraints so that your changes will persist across academic quarters even as the web is regenerated quarterly.

The per-course webs are edited using the CSE Content Creation and Maintenance Tool, and as such have “orthodox” design elements as described in that tool.

The elements that can be edited are:

\textbf{upper-left thematic image}
This image can be changed to any of the standard images supported by the content tool. An easy way to do that is to use the tool to edit the existing file, select a new image, save, then copy the modified file from \texttt{/cse/www/tmp/} to the course web directory. You can use other image files, but unless it’s one of the standard images, it will revert to the default image during the quarter break. Don’t like our images? We are happy to install your 133x100 image in the content tool.

\textbf{titled heading}
The course title is obtained from the online course catalog and used to build the title in the heading and browser frame. Except for courses not listed in
the catalog, notably 590s, changes will be lost. In the case of courses not in the catalog, the title is inherited from the per-course web, so changes will be preserved.

**navigation bar**
This is orthodox content that should never be changed, and changes will be discarded at quarter break.

**lefthand navigation**
You can add your own lefthand navigation using the content tool. In the example image, the “Related Courses” section was added using the content tool. Guidelines: you must introduce your lefthand navigation with a heading (such as “Related Courses”), and you should place it after the standard sections: “Current Quarter,” “Previous Quarters,” “Prerequisites,” and “Standard Syllabus” (some of which will not be present for some courses). If you don’t place it after standard sections of navigation, it will be preserved but migrate there next time the per-course index is generated.

**HTML content pane**
You can add content to the HTML content pane only between two special comments in the file:

```html
<!-- Custom HTML follows.  
See http://www.cs.washington.edu/lab/course-computing/percourse.html-->
....
<!-- End of custom HTML. -->
```

Use a text editor or use the content tool, but avoid fancy HTML editors, which add and subtract code in a manner sure to confuse the content tool.

**footer**
This is orthodox content that should never be changed, and changes will be discarded at quarter break.

By following these guidelines, the content you add to per-course webs will persist.

### A.1.3 Support for CSE590 Webs

Certain courses, mostly sections of CSE 590, *Special Topics in Computer Sciences*, are not listed in the online course catalog. These fall into two broad categories:

1. those that are taught once or twice or have no cohesive theme
2. those that are taught frequently and have a persistent theme

It is useful to provide a per-course web for those courses in the latter category, but the information we need can’t be gleaned from the online catalog. To help create and maintain them, there exists a **Per-course Web Configuration** tool.

The capabilities of the tool are easily seen by running it. The basic idea is that it creates and maintains a simple per-course configuration file that establishes similar information as is provided by the online catalog for standard courses.
Because the information in the file it creates is used in the creation of any per-course web, and overrides information from both the existing per-course web and the online catalog, it can be useful in unusual circumstances for cataloged courses as well:

- for correcting or expanding upon information from the catalog
- for limiting the depth of the "previous quarters" section of a web to a user-specified value

This tool is expected to be used primarily by the webmaster, but is available to any instructor.

### A.2 Creating and Setting Ownership and Permission of Project Directories

#### A.2.1 Project Directories Based on Project Groups

Assume the project directory in question is `/projects/instr/04sp/cse000`, and that you have requested 15 project groups. The project group names for a given course all have the same prefix, which identifies that course, either `cse000x`, where $x = a, b, c, ...$; or, for a longer course name such as `cse000zz`, `c000zz-x`. We recommend you create group directories with names `a, b, c, ...`, which correspond to the tail of the group names, e.g.:

```
  cd /projects/instr/04sp/cse000
  mkdir a b c d e f g h i j k l m n o
```

After the directories are created, you will need to set group ownership and mode. Setting group ownership has proved to be tricky for TAs in the past, so the process has been streamlined by providing a script to do it for you. However, the script will only work if a valid group name is produced when you prefix a given string to each directory name, e.g. in this case prefix "cse000" + directory "a" yields "cse000a", the group name you want to assign to this directory. The script takes a single parameter, viz. the prefix to which directory names will be appended to form group names. For example:

```
  cd /projects/instr/04sp/cse000
  /cse/courses/bin/setprojdirs cse000
```

will set group ownership and mode for all of the group directories under `/projects/instr/04sp/cse000`. Enter `/cse/courses/bin/setprojdirs` (no parameters) for more information.

#### A.2.2 Project Directories Based on CSENetID

For some courses, e.g. `cse331`, individual project directories are set up, each named with the CSENetID of a student. In addition, the directories might be populated with some initial information, e.g. a repository of code to be modified by the student. The problem here is making sure each student owns that directory and everything in it. Only administrative
accounts can change file and directory ownership, so teaching staff either need to notify support@cs or do it themselves by following this procedure:

1. Set up the student directories, e.g. for student ‘jostudent’ in course cse331 set up by TA ‘jota’ this might be how things looked after setting up the directories:

   ```
   % pwd
   /projects/instr/11wi/cse331
   % ls -l jostudent
   total 12
   drwxrwx--- 3 jota cse331 4096 Jan 3 20:13 ./
   drwxrwxr-x 57 instructor cse331 4096 Jan 30 15:54 ../
   drwxrwx--- 6 jota cse331 4096 Jan 3 20:13 REPOS/
   ```

2. In the course directory create the file ‘run.change.owner.dirname’, e.g.:

   ```
   touch /projects/instr/11wi/cse331/run.change.owner.dirname
   ```

   Within a short time (minutes) a process on the projects fileserver will run that does the following:

   ```
   % cd /projects/instr/11wi/cse331
   % for i in *; do [[ -d $i ]] && chown -R $i $i; done
   ```

   i.e. recursively changes the ownership of all directories to the name of the directory. After this happens one would see:

   ```
   % pwd
   /projects/instr/11wi/cse331
   % ls -l jostudent
   total 12
   drwxrwx--- 3 jostudent cse331 4096 Jan 3 20:13 ./
   drwxrwxr-x 57 instructor cse331 4096 Jan 30 15:54 ../
   drwxrwx--- 6 jostudent cse331 4096 Jan 3 20:13 REPOS/
   ```

### A.2.3 Ensuring Course Group Access

After setting things up it may happen that group read-write access, or even the group itself, can be changed by the student. To re-establish course group access, the process is similar to the previous section, only this time the file to touch is ‘run.change.group.perms’, e.g.

for the project directory example of the previous section:

```
   touch /projects/instr/11wi/cse331/run.change.group.perms
   ```

and, in effect, the following will be done within a few minutes

```
cd /projects/instr/11wi/cse331
for i in *
do
  [[ -d $i ]] || continue
  chgrp -R 'groupof $i' $i
  chmod -R g+rwX $i
done

where `groupof` returns the group of its argument, a directory.

**A.3 Clearing Out Course Groups**

Sometimes course groups are left with a membership that reflects the projects of a previous quarter. The `grpadmin` command has no “zapgroups” option, but a `zapgroups` script to remove all members from a group has been provided. Enter `/cse/courses/bin/zapgroups` (no parameters) for details. For example, if you wanted to clear out all the course groups mentioned above, you could enter the following command:

```
/cse/courses/bin/zapgroups 'perl -e 'for("a".."o")print "cse000$\n"')'
```
# Table of Contents

1 **Introduction** .......................................................... 1

2 **Hardware and Software Facilities** ......................... 1
   2.1 Instructional Labs and Servers ............................... 1
   2.2 Instructional Workstation Testbed via Remote Desktop .... 2
   2.3 Quarterly Course Computing Requests - Response Required .... 2
   2.4 Free Microsoft Software for Students ........................ 2

3 **Groups and Accounts for Instructors, TAs, and Students** ....... 2
   3.1 Groups ............................................................... 2
   3.2 Accounts ............................................................. 3

4 **Course Files** ......................................................... 4
   4.1 Naming Conventions ............................................. 4
   4.2 General Purpose Files .......................................... 5
   4.3 Web Documents .................................................... 6
      4.3.1 Course Home Pages ........................................ 6
      4.3.2 Course WIKIs ............................................... 7
   4.4 Student Project Areas and Repositories for Version Control
      Systems ................................................................. 8
   4.5 Archives ............................................................. 9
      4.5.1 General Purpose Archives ................................. 9
      4.5.2 Course Web Archives ...................................... 9
      4.5.3 Student Project Archives ................................. 9
      4.5.4 Archive Permissions ...................................... 9

5 **Course Email Lists** ................................................ 10

Appendix A **Appendix** .................................................. 11
   A.1 Specialized Course Web Access ............................... 12
      A.1.1 Course Web Access for Undergrad and Non-Major TAs ... 12
      A.1.2 Editing Per-Course Webs .................................. 12
      A.1.3 Support for CSE590 Webs .................................. 13
   A.2 Creating and Setting Ownership and Permission of Project
      Directories ............................................................ 14
      A.2.1 Project Directories Based on Project Groups .......... 14
      A.2.2 Project Directories Based on CSENetID ................. 14
      A.2.3 Ensuring Course Group Access ............................ 15
   A.3 Clearing Out Course Groups ................................... 16

............................................................. 17