

# 5

## Enabling your Moodle Site to Function as an Information Portal

*Our Moodle site is truly starting to take shape as a CIMS now that we have created course categories, created and enrolled teachers in courses in each of our categories, created student accounts and enrolled students in courses, and finally, set up our site such that it will support and conform to standards adhered to by our institution. It is time now to take a step back from the educational administrator's or teacher's perspective and look at our site from the perspective of the students who will be interacting with it.*

*Most educational programs present students or, in the case of very young students, their parents or guardians with detailed information about the programs offered. Likewise, most educational programs that are made up of a series of courses almost always deliver those courses and their curriculum, in a somewhat regulated fashion. For example, a student is not allowed to take an advanced math class prior to completing an entry-level course or series of courses. Additionally, many programs that organize their curriculum such that students work toward obtaining certificates and degrees, also provide information to students about required classes they must take, and how many elective courses they may select freely as well as from which category or categories the electives may be chosen.*

*With minor modification, Moodle can serve as an information portal allowing an institution to efficiently deliver detailed information about its faculty, programs, and curricula to students and their families.*

In this chapter, we will explore some of these possibilities through discussion of and experimentation with the following:

- ◆ Modifying display settings
- ◆ Removing course lists from the category page
- ◆ Increasing the detailed summary settings
- ◆ Increasing the number of courses displayed in a user's profile
- ◆ Expanding content display possibilities
- ◆ Installation and use of the Content Pages block
- ◆ Customization of third party themes

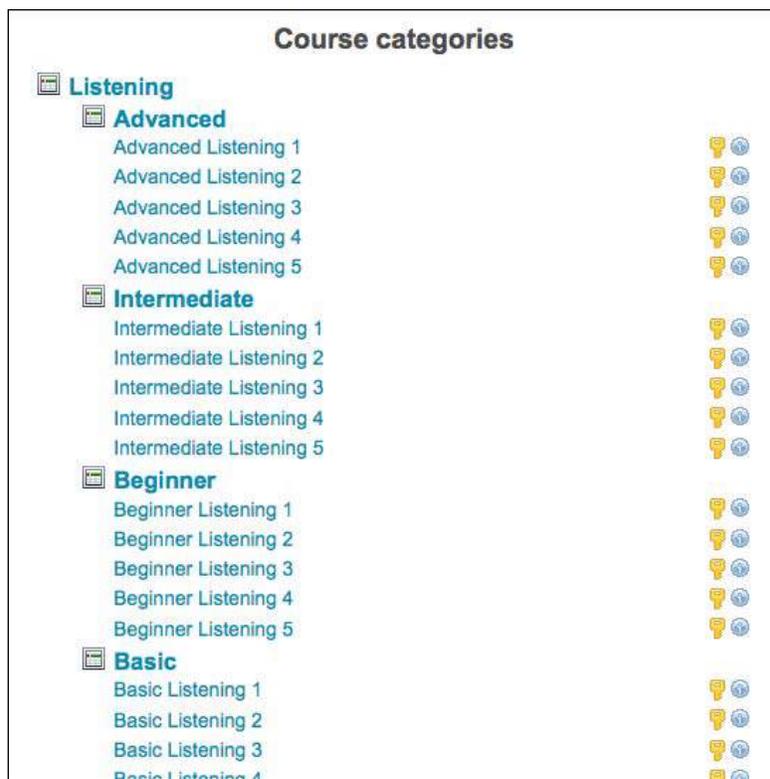
## **Modifying display settings**

As we have learned so far, Moodle is a very powerful and flexible tool for setting up a Virtual Learning Environment (VLE). As we mold our Moodle site into a CIMS however, we might stumble into a few settings, written into Moodle's core, that we'd like to 'tweak' a little to better enable us to use Moodle to help us manage our curriculum and the information that surrounds it.

### **Time for action – removing course lists from the category page**

The first of these such 'tweaks' is to change how the **Course category** page is displayed. It is important to note that this will involve making a small change to the core Moodle code. This will mean that when you upgrade your Moodle instance to a newer version, you will need to make this adjustment again in the new version. Ideally, you should keep adjustments like this, sometimes called hacks, to a minimum as they can complicate the upgrade process. Always keep a record of the changes you make so that you can easily transfer them to your new instance of Moodle when you upgrade. Additionally, if your adjustment involves a feature that you feel should really be a part of the Moodle core complete with an adjustable setting via the Moodle browser interface, please make the request via the Moodle Tracker found at <http://tracker.moodle.org/secure/Dashboard.jspa>.

To see the page display that we are going to modify, navigate to the following page of the site, that we have built up to this point, found at <http://yoursite.com/moodle19/course/index.php> and you will see a screen similar to the following screenshot:



You'll notice, as you scroll down the window, that this list is rather cumbersome due to the fact that all of the courses in each category are being displayed on this page. For our program, we want to display only the categories on this page so we will change the Moodle core settings to accomplish this task through a very simple edit in one Moodle file. Follow these steps to make the change:

1. If you don't already have a text editing program that you are comfortable with, download and install one from the internet. Here are a couple of open source recommendations:
  - Mac OS: TextWrangler (<http://www.barebones.com/products/>)
  - Windows: Notepad++ (<http://notepad-plus-plus.org/>)
2. Open your text editing program and with it then open the `lib.php` file located at the following path: `yourmoodledirectory/course/lib.php`.

3. On or around line 17, you will see the following PHP code:

```
if (!defined('FRONTPAGECOURSELIMIT')) {  
define('FRONTPAGECOURSELIMIT', 200); // maximum number of  
courses displayed on the frontpage
```

4. Change the 200 to a 1 and save the file.
5. Now refresh the page at <http://yoursite.com/moodle19/course/index.php> and you will see the course listings disappear, as shown in the following screenshot:



Course categories	
Listening	
Advanced	5
Intermediate	5
Beginner	5
Basic	5
Speaking	
Advanced	5
Intermediate	5
Beginner	5
Basic	5
Reading	
Advanced	5
Intermediate	5
Beginner	5
Basic	5
Writing	
Advanced	5
Intermediate	5
Beginner	5
Basic	5
Miscellaneous	4

### ***What just happened?***

By changing the 200 to a 1 in the `lib.php` file, we have modified Moodle to display categories only when there is more than one course on the site. In addition to keeping a record of this change somewhere outside of your Moodle site, you may also want to place a comment in the code file using two forward slashes (`//`), as you see on the line containing the setting we changed. For example, you could add a quick explanation behind the maximum number of courses line to explain that you changed the setting to 1 from the default value of 200. This comment might look like the following code:

```
CODE // maximum number of courses *changed from 200 to 1
```

Note that the actual code has been replaced with the term `CODE` in order to keep the text on the same line.

## Time for action – increasing the detailed summary setting

Now that we have modified the course category display setting, the categories, and the number of courses they contain, are the only things that can be seen from the **Course category** page. Clicking on a category will display the courses contained within, as shown in the following screenshot:



The example provided here is ideal as the listing for each course can contain detailed information about the course, such as the days and times it meets, the location, information about the syllabus, the name and even possibly a picture of the instructor if it is entered through the HTML editor portion of the description setting from within the course. However, a problem arises when we have more than ten courses in a category. In a default Moodle installation, Moodle is set to abbreviate the course descriptions when there are more than ten on a page. The following screenshot shows this abbreviation style with just five courses (this was accomplished by changing a core setting that we will edit shortly):



This style is much more efficient in terms of screen space but it does not provide students, who might be considering registering for a course, with enough information about the course, its requirements, the syllabus, the instructor, and so on. To obtain this information, the student must click on **i** for "information" and view the details in a pop-up window. The need to do so for multiple courses can be quite tedious from the perspective of the student who is browsing the curriculum and designing his or her schedule. We will therefore change the default setting of ten courses per category to a larger number to ensure that the detailed summaries are displayed for all courses when listed inside their categories. To do so, follow these steps:

1. Open your text editor and use it to open the same file that we edited for the previous task found at this path: `yourmoodledirectory/course/lib.php`.
2. On or around line 9, you will see the following code:  

```
define('COURSE_MAX_SUMMARIES_PER_PAGE', 10); // courses
```
3. Change the `10` to a number high enough to accommodate the maximum number of courses you have in a category. We'll make this `40`. This far exceeds the number of courses we have, or can conceive of, in categories and thus should be more than sufficient.
4. Refresh your screen and, if you had more than 10 courses in a category, you will see a change in how they were displayed. If you are following the tutorial outlined in this book, you will only have five courses per category. To see the abbreviated display style as shown in the previous screenshot, you can change the setting to `1`, save the file, and then refresh your screen. Just don't forget to change it back to a larger number once you have finished experimenting with the setting.

### ***What just happened?***

By changing one number in one line of code, we have modified how courses are displayed within categories in order to ensure that more detail is displayed with each course. For an institution that wishes to use Moodle as the primary course registration system, this allows students more comprehensive access to courses without having to view a pop-up window for each course.

### **Pop quiz**

1. Why did we change the setting that controls how many course summaries are listed on a page?
  - a. Because we want the list to be as brief as possible.
  - b. Because we want the course list to include the instructor and summary information.

- c. Because the abbreviated list shows more information than we really need.
- d. Because we weren't sure how many courses we would have in each category that we create.

## Time for action – increasing the number of courses displayed in a user's profile

The last setting that we will modify for now is the number of courses that are displayed in a user's profile. When you click on a user's name in Moodle, your browser is redirected to the profile page for that user where various details, including the courses the user is enrolled in, are displayed. As a default setting, Moodle will only display 20 courses for a user. If the number of courses a user is enrolled in has exceeded 20, the course list for that user will end with three dots, as shown in the following screenshot. Note, this screenshot shows only one course followed by the three dots. This is a result of the setting we are about to change being set to 1 for the purpose of demonstration.



Change the setting from the default of 20 to a higher number by following these steps:

1. Open the following file with your text editor: `yourmoodledirectory/user/view.php`.
2. Look for the following line of code found on or around line 356:  

```
if ($shown==20) {
```

3. Change the 20 to a higher number such as 40.
4. Save the file and refresh the screen containing a user profile to see the results. If you are following the tutorial outlined in this book, each teacher will only be teaching four classes so the display will not be affected. To see the abbreviated list display style, as shown in the previous screenshot, you can change the setting to 1, save the file, and then refresh your screen. Just don't forget to change it back to a larger number once you have finished experimenting with the setting.

### ***What just happened?***

By again changing just one number in one line of code, we changed the setting that governs how many courses are displayed on a user's profile page. The default setting is 20 so we changed the setting to 40 to make sure that all courses will be displayed. This may seem a bit strange, as it is probably unlikely that a user would be teaching or taking more than 20 courses at a time. The reason for wanting to be able to see all courses a user is enrolled in that we will discuss in detail in a later chapter, is that we will be making courses invisible but keeping their enrollment intact and want to make sure that all of the enrollment information is displayed to the site admin via the user profile page.

## **Expanding content display possibilities**

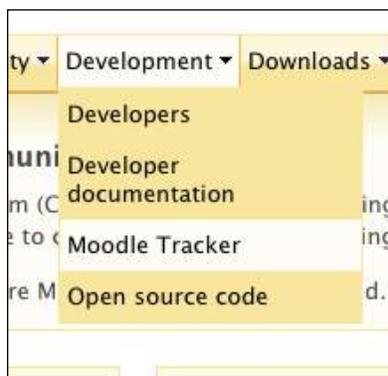
In addition to customizing how Moodle displays course and category information, we want to be able to display information that is not directly linked to any single course. Information about degree or certificate requirements, as mentioned earlier, for example, is something that is frequently provided to students and stakeholders by educational institutions. This information can be included in our Moodle site via the use of a contributed block called Content Pages.

### **Time for action – installation of Content Pages block**

To install the Content Pages block, which will allow us to create web pages that can be accessed by any user viewing our site, whether they are registered users or not, follow these steps:

1. The Content Pages block is not currently a block that is maintained on the Moodle **Modules and Plugins** page although, it may soon be added. First check the **Modules and Plugins** page for the block to see if it is available.
2. If the Content Pages block can be found in the Modules and Plugins database, you should download the most recent package and follow the installation process from step 6 below.

3. If the block cannot be found on the **Modules and Plugins** page, you will need to navigate to the Moodle Tracker site, which can be found from the front page of <http://www.moodle.org> by moving your mouse over the **Development** drop-down menu and then clicking on **Moodle Tracker**, as shown in the following screenshot:

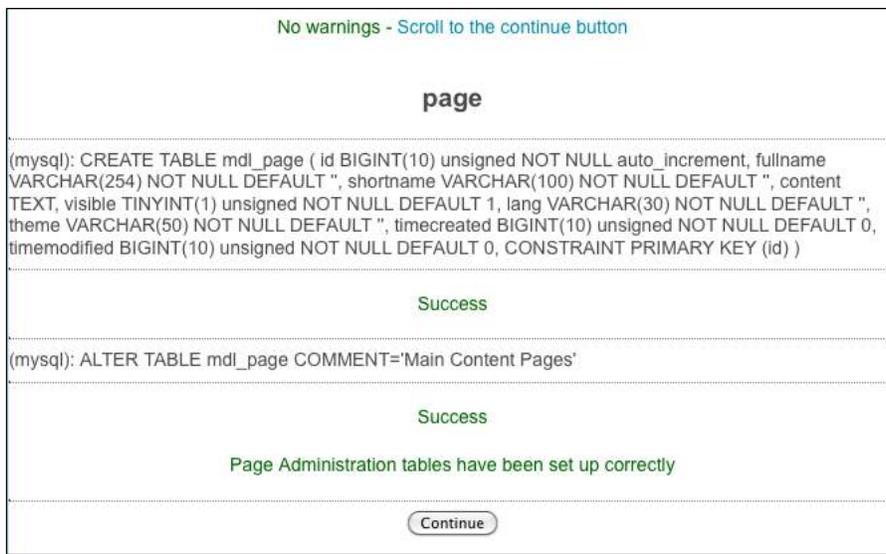


4. From the front page of the Moodle Tracker enter `CONTRIB-1303` in the **Quick Search** field, as shown in the next screenshot, and press enter/return to initiate the search:



5. The result will be a Tracker record for the Content Pages block. The package to download will be the zipped file. As this book goes to press, the newest version is `moodle195-pages-SENT_TRACK-CONTRIB-1303-26Jan2010.zip`. Click on the newest package to download it to your computer.
6. Decompress the zipped package and note that the contents include a folder for each of the following directories:
  - admin
  - blocks
  - lib
7. The admin folder will contain two subdirectories or folders (**roles** and **settings**), each containing PHP files. Locate the target directory/folder in your Moodle directory and first move the file already in your Moodle directory with the same name to a backup location such as your desktop. Then move the contents of each subdirectory from the unzipped package to the appropriate location in your Moodle directory.

8. For the **blocks** folder, there will not be a **page** folder in your **block** directory, so you will only need to copy the contents to the **block** directory in your Moodle directory.
9. Repeat step 8 for the **lib** folder making sure to first copy the existing files to a backup location.
10. Once you have moved all of the Content Pages block components into their appropriate locations, navigate to the front page of your Moodle site logged in as a user with administrative privileges and click on the **Notifications** link. This will initiate the installation process, which entails the creation and set up of the database table that will be used by this block, as shown in the following screenshot:



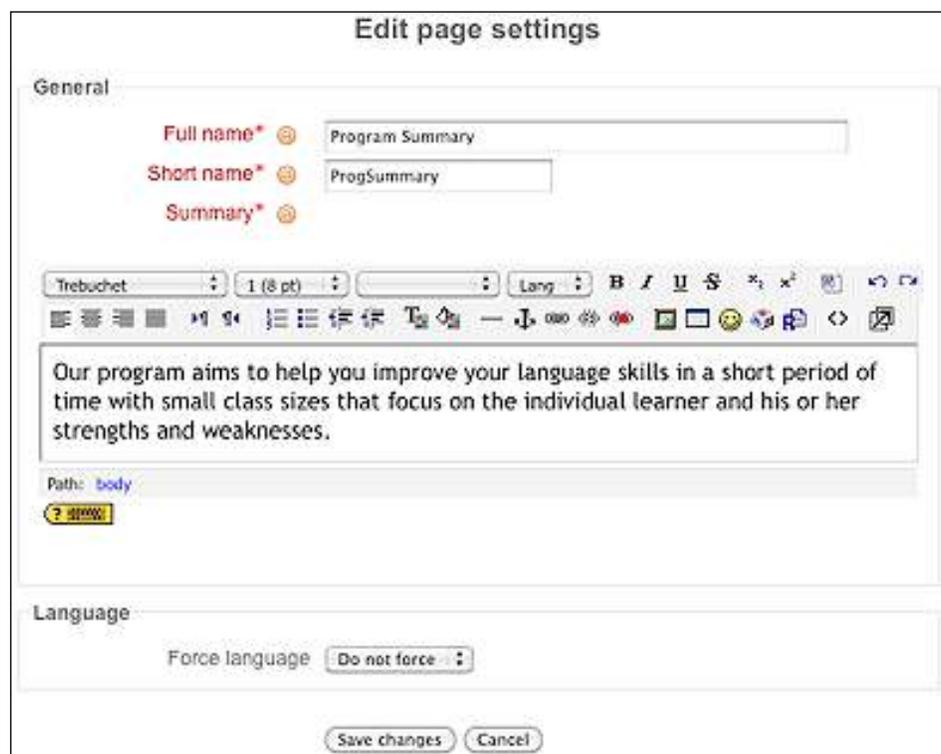
11. Click on the **Continue** button to finish the process and then on the top-left breadcrumb to return to the front page of your site. You will notice a new link in the **Site Administration** block between the **Courses** and **Grades** folders titled **Add/edit pages**, as shown in the following screenshot:



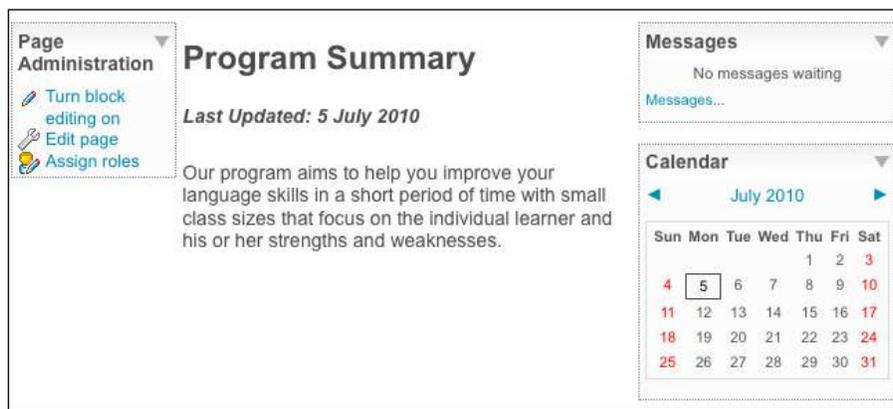
- 12.** Click on the **Add/edit pages** link to view the Content Pages block interface. As we have not yet created any pages with the block, the list will be empty and only a page title and an **Add a new page** button will appear, as shown in the following screenshot:



- 13.** Click on the **Add a new page** button to create a new page. For this example, we will create a page that summarizes our program and its focus. The **Full name** will appear at the top of the page you are creating, so make it something that is suitable as a title. The **Short name** is what will be used in the breadcrumb path so use something that is short but explanatory. Our example is shown in the following screenshot:

A screenshot of the "Edit page settings" form. The form is divided into sections: "General" and "Language". In the "General" section, there are three required fields: "Full name\*" with the value "Program Summary", "Short name\*" with the value "ProgSummary", and "Summary\*" which is empty. Below these fields is a rich text editor with a toolbar showing font settings (Trebuchet, 1 (8 pt)) and various editing tools. The text in the editor reads: "Our program aims to help you improve your language skills in a short period of time with small class sizes that focus on the individual learner and his or her strengths and weaknesses." Below the text editor, the path is shown as "body" with a "show" button. In the "Language" section, there is a "Force language" dropdown menu set to "Do not force". At the bottom of the form are "Save changes" and "Cancel" buttons.

- 14.** Click on the **Save changes** button to create and view the page. Your page will look similar to the following screenshot:



- 15.** If you copy the address that appears in your browser window, log out of your site, and point your browser to that address, you will see that the **Page Administration** block is no longer visible but everything else appears the same.

### ***What just happened?***

We have just installed the Content Pages block, which included adding or, in the case of some files, overwriting some files to the Moodle directory. Next we finished the installation process by clicking on the **Notifications** link found in the **Site Administration** block from the front page of our site, which triggered Moodle to install a new table in the database that is used to store page information. Finally, we created a simple test page using the newly installed Content Pages block. The Content Pages block will allow us to create various web pages that can be accessed from the menu bar of the theme that we installed in *Chapter 1, Welcome to Moodle as a Curriculum and Information Management System (CIMS)*! Additionally, web pages can, of course, be added and nested within pages that are linked from the menu bar via the use of links. As mentioned earlier in this chapter, this enables our Moodle site to organize and display information about our educational program to students and other stakeholders such as parents and guardians.

### **Time for action – customizing the Content Pages block**

Although it is not necessary, for the purpose of esthetics, and to again demonstrate how easy it is to customize your site, we are now going to remove the last updated line, center the title, and remove the blocks displayed on the right side of the page. Follow these steps to make these additional changes:

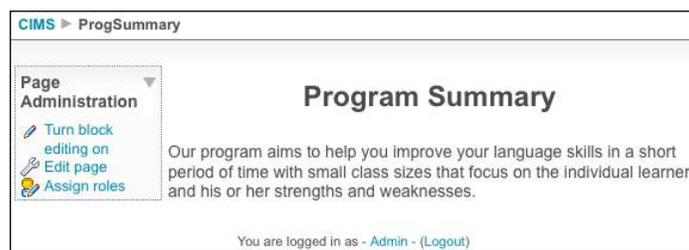
1. With your text editor, open the file found at the following path:  
yourmoodledirectory/blocks/page/view.php and comment out the line found on or around line 116. The line is:
 

```
print_heading("<i>Last Updated: ".userdate($page->timemodified,
get_string('strftimedate'))."</i>", 'left', 4);
```
2. It can be "commented out" by placing two forward slashes at the beginning of the line. After you have added the slashes, save the file and refresh your browser window. You will see that the **Last Updated** message has disappeared.
3. Next, in the same file, on the line just above the line we just commented out, which should be on or around line 115, change the alignment of the heading from left to center, as shown in the following code:
 

```
print_heading($page->fullname, 'center', 1);
```
4. Save the file and refresh your browser window and you will see the heading move to the center of the page.
5. Finally, to remove the blocks shown on the right side of the page, we will need to comment out a small portion of the code found in this same file. Add double forward slashes to the portion used to format the right column of the page found from around lines 124 to 132. The commented out code will look like this:

```
// The right column
//if (blocks_have_content($pageblocks, BLOCK_POS_RIGHT) ||
$editing || $PAGE->user_allowed_editing()) {
//  echo '<td id="right-column">';
//  print_container_start();
//  blocks_print_group($PAGE, $pageblocks, BLOCK_POS_RIGHT);
//  print_container_end();
//  echo '</td>';
//}
//  break;
```

6. Save the file again and refresh your browser window and you will see the message and calendar blocks on the right side of the page disappear. The page we just created should now look like the following screenshot:



7. Note again here that if you copy the address from your browser address bar, log out of your site, and view the page, the **Page Administration** block does not appear. The administration block is only viewable by the site administrator and will not clutter the window for students and other users.

### ***What just happened?***

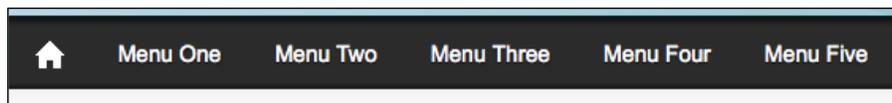
In order to modify the display settings for the pages we create with the Content Pages block, we opened and edited the `view.php` file found in the `page` directory located in the `blocks` directory that is found inside the `Moodle` directory. The simple modifications we made were to remove a line that provides information about when the page was last edited, as well as to prevent blocks from being displayed on the right side of the screen by commenting out the portion of the code that performs this function. Additionally, we changed a formatting setting to center the page heading instead of having it aligned on the left side of the page. These changes are all very simple and require no detailed knowledge of PHP.

### **Have a go hero**

In the next section of this chapter, we are going to walk through the process of customizing our third-party theme and part of this customization process will require some web pages to link to. So before you move on, take a minute to create seven more pages using the Content Pages block. These pages will be linked to the submenu links that appear when you move your mouse over the **Menu One** title in the menu bar of our Aardvark theme. Get a little creative and add some pictures or graphics to experiment with the types of pages you might create with this block.

## **Customization of third party themes**

In chapter one, we installed a third party theme called Aardvark. One of the reasons for using this theme, as opposed to the default Moodle theme, will now become more apparent as we link some pages to links provided by the drop-down menus that appear on the front page of our site. We will start by customizing the five menu items that appear at the top of the page, as shown in the following screenshot:

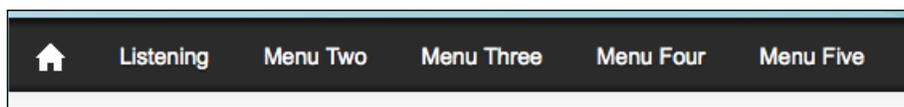


## Time for action – customizing the menu bar

To change the titles of the menu items shown in the menu bar, follow these steps:

1. With your text editor, open the file found at the following path:  
yourmoodledirectory/theme/aardvark\_pro/aardvark\_menu.php.
2. On or around line 43, change the text **Menu One** to **Listening**, as shown in the following code:  

```
<li><div><a href="<?php echo $CFG->wwwroot.'/ ' ?>">Listening</a>
```
3. Save the file and refresh your browser window. You will see **Listening** take the place of the **Menu One** title, just as it appears in the following screenshot:



### Using foreign language fonts in the menu bar

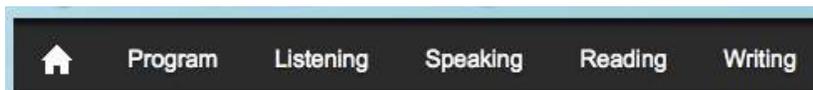


If you want to use foreign language menus that contain letters or characters outside of the Latin alphabet, you will need to save the `aardvark_pro.php` file using the UTF-8 character encoding method. Additionally, you'll want to choose the option **No BOM** when you save the file. You may need to experiment with the **BOM (Bit Order Mark)** setting to obtain the result you desire. In the case of Japanese language fonts, for example, saving the file without **No BOM** selected may result in the text for the menus moving down and outside of the black background, as shown in the following screenshot. With TextWrangler, this can be accomplished by clicking on **Options** from within the **Save As** window and selecting the **No BOM** option.



4. Repeat this process for all of the menu titles. Each menu title appears in a line of code that looks like the one provided in step 2. The locations are on or around the following lines in the file:
  - ❑ **Menu Two:** Line 64
  - ❑ **Menu Three:** Line 85
  - ❑ **Menu Four:** Line 106
  - ❑ **Menu Five:** Line 126

5. Once you have changed all of the menu titles, save the file and refresh your browser window to see the changes take effect. Our example looks like the following screenshot (note that, I decided to move the **Listening** menu title to the right to make room for a **Program** link):

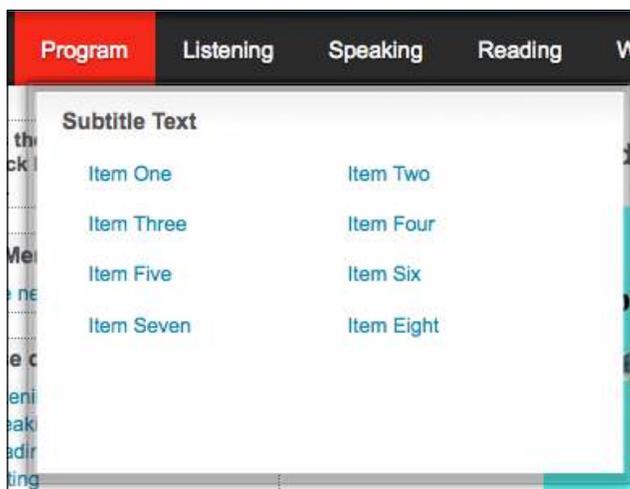


### ***What just happened?***

We just modified the `aardvark_menu.php` file in order to add more descriptive menu titles to the menu bar of our theme. This simple process allows us to categorize each list of links that appear when visitors to the site move their mouse over each menu item.

### **Time for action – customizing the submenus**

Now that we have customized the menu titles, we need to continue by customizing the titles that appear in each box that shows up when we move our mouse over the title. We also need to make sure that each link, the menu title as well as the items that appear in the boxes, points to a page with relevant information. For our example, we will work with the **Program** menu item and its submenu. In its current state, it looks like the following screenshot:

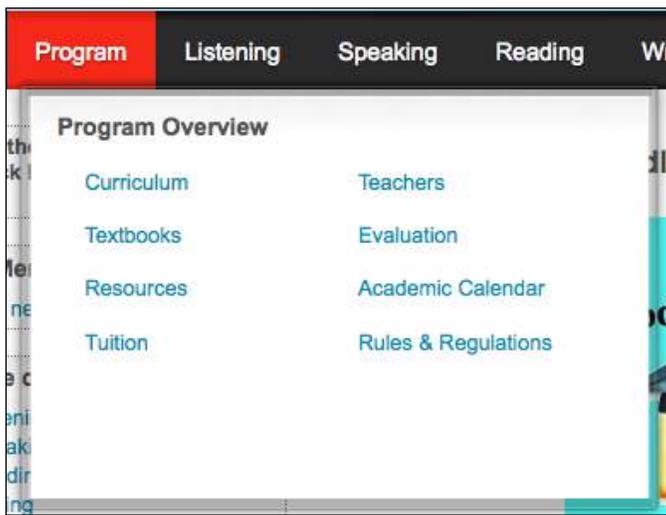


To change the title that appears at the top of the window, as well as the text for the eight items, follow these steps:

1. With your text editor, open the `yourmoodledirectory/theme/aardvark_pro/aardvark_menu.php` file to change the menu bar titles.
2. The text we need to edit can be found between lines 46 and 57. I will change the titles for our example as follows (note that, the line numbers listed are approximate and may vary depending on the version of the file):
  - Line 46: **Subtitle Text: Program Overview**
  - Line 50: **Item One: Curriculum**
  - Line 51: **Item Two: Teachers**
  - Line 52: **Item Three: Textbooks**
  - Line 53: **Item Four: Evaluation**
  - Line 54: **Item Five: Resources**
  - Line 55: **Item Six: Academic Calendar**
  - Line 56: **Item Seven: Tuition**
  - Line 57: **Item Eight: Rules & Regulations**
3. The following is a copy of the lines of code from the `aardvark_menu.php` file after being edited for our example:

```
<h4>Program Overview</h4>
<?php
    $text = '<li><a href="">Curriculum</a></li>';
    $text .= '<li><a href="">Teachers</a></li>';
    $text .= '<li><a href="">Textbooks</a></li>';
    $text .= '<li><a href="">Evaluation</a></li>';
    $text .= '<li><a href="">Resources</a></li>';
    $text .= '<li><a href="">Academic Calendar</a></li>';
    $text .= '<li><a href="">Tuition</a></li>';
    $text .= '<li><a href="">Rules & Regulations</a></li>';
```

4. After changing the titles, save the file and refresh your browser window. Move your mouse over the **Program** menu item to activate the submenu window and observe the changes that have been made. The following is a screenshot of our example:



### ***What just happened?***

We have just changed the titles of the submenu items that appear when you move your mouse over the **Program** heading, which we changed earlier. For our example program, these submenu links will lead us to pages that we created with the Content Pages block. This will allow us to provide access to various types of information about our educational program from the front page of our Moodle site.

### **Time for action – adding links to the submenu items**

The final step in this process is to add link destinations for all of the menu items that appear as links. This includes the eight items that appear in the submenu window as well as the **Program** and other menu items in the menu bar. Links to pages we have created with the Content Pages block as well as to any other location in your Moodle site or to an external site are possible. To insert the links, follow these steps:

1. With your text editor, again open the file we just edited found at `yourmoodledirectory/theme/aardvark_pro/aardvark_menu.php`.

- 2.** Add links to the code from approximately lines 43 to 57 as shown in the following code snippet. For the submenu items, adding a link is as simple as copying and pasting a link between the quotation marks that appear after the `href=` code. One simple way to copy a link is by navigating to the desired page with your browser and copying the link from the address bar. To add a link to the menu item, remove all of the code found between the quotation marks and enter your link. Here is the code after adding all of our links. (Note, I have used only links to pages created via the Content Pages block but you can add any link here. Also note that the address contains `//localhost:8888/moodle19...` This is specific to the Mac OS MAMP package. If you are using XAMPP on Mac or Windows, the address will not contain 8888.)

```
<li><div><a href="http://localhost:8888/moodle19/blocks/page/view.php?id=1">Program</a>
  <ul>
    <h4>Program Overview</h4>
    <?php
$text .= '<li><a href="http://localhost:8888/moodle19/blocks/page/view.php?id=2">Curriculum</a></li>';
$text .= '<li><a href="http://localhost:8888/moodle19/blocks/page/view.php?id=6">Teachers</a></li>';
$text .= '<li><a href="http://localhost:8888/moodle19/blocks/page/view.php?id=3">Textbooks</a></li>';
$text .= '<li><a href="http://localhost:8888/moodle19/blocks/page/view.php?id=7">Evaluation</a></li>';
$text .= '<li><a href="http://localhost:8888/moodle19/blocks/page/view.php?id=4">Resources</a></li>';
$text .= '<li><a href="http://localhost:8888/moodle19/blocks/page/view.php?id=8">Academic Calendar</a></li>';
$text .= '<li><a href="http://localhost:8888/moodle19/blocks/page/view.php?id=5">Tuition</a></li>';
$text .= '<li><a href="http://localhost:8888/moodle19/blocks/page/view.php?id=9">Rules & Regulations</a></li>';
```

- 3.** Once you have entered all of the links, save the file and refresh your browser window. Move your mouse over the first menu item and click on one of the links to verify that the links are functioning as desired.

- Note, the links entered in the previous code are from a series of pages created via the Content Pages block. Viewing the pages from the Content Pages administration page, which is accessible via the **Add/edit pages** link found in the **Site Administration** block, will provide you with a quick and easy way to access each page and copy its link from your browser address bar so you can then paste it into the `aardvark_menu.php` file. Our page list appears, as shown in the following screenshot:



### ***What just happened?***

We have just added link destinations to all of the submenu items that appear under the **Program** heading in the menu bar of our Aardvark theme. For our example, we used pages that were created by the Content Pages block but you could, of course, use any web page address desired.

## **Time for action – adjusting the display style of the Content Pages admin block**

It is necessary to make note here of one additional small adjustment that should be made to correct a minor block formatting issue that appeared when we removed the blocks from the right side of the page by editing a file in the Content Pages block package. Under the following conditions, you will notice that when viewing a page created by the Content Pages block, the **Page Administration** block is being resized to take up half of the width in your page:

- You are logged in as admin.

2. The information you have entered in the page does not take up much horizontal space.
3. You are using the Aardvark theme (note, depending on how they are styled, other themes may have the same display issue).
4. The following screenshot demonstrates this formatting issue and, while it is only visible when logged in as admin, it is nice to fix it as, in some cases, it will better enable you to see what your page is going to look like to other users without having to logout of your site and view the page:



5. To correct this display issue, follow these simple steps:
  - With your text editor, open the file located at the following path: /yourmoodledirectory/theme/aardvark\_pro/aardvark.css.
  - On or around lines 62 to 65, you will see settings that control how the left column is formatted. Add a line containing width:150px; after the following lines:
 

```
#layout-table #left-column{
  vertical-align:top;
  padding-left:4px;
  padding-right:0px;
```
6. Save the file and refresh your browser window while viewing a page created with the Content Pages block. If you were experiencing the issue shown in the previous screenshot, you will see the **Page Administration** block being resized to roughly the width of the longest item it contains, as shown in the following screenshot. If you prefer to have the **Page Administration** title on one line, increase the width pixel setting until you achieve the desired result.



## **What just happened?**

In order to make sure that the width of the **Page Administration** block is constant and does not expand to take up a large portion of the window, we also made a small adjustment to the CSS (Cascading Style Sheet) file in our theme in order to control the size of the **Page Administration** block. This fixed a minor cosmetic issue that was apparent when we logged into our site as a user with administrative permissions and viewed a page that we created with the Content Pages block that did not have a large amount of text.

## **Have a go hero**

Now it's your turn! Plan out a scheme for the types of information that you'd like to have accessible from the front page of your Moodle site. Make pages with the Content Pages block and link them to the menu items available in the menu bar. Provide links to courses in your Moodle site or to other websites hosted by your institution or to other websites relevant to your program. As you implement this plan using the menu bar and Content Pages block, your Moodle site will begin to stand even more strongly as a CIMS.

## **Time for action – customizing the logo**

One last thing that we should customize before finishing up this chapter is the logo that appears at the top of the front page. In *Chapter 1, Welcome to Moodle as a Curriculum and Information Management System (CIMS)!*, we added a graphic to the topic section that appears at the top of the front page, but we may want to use that area for other purposes and as there is already space at the top of the page allocated for a logo, it only makes sense to add our school or program logo there. To change the Aardvark logo, follow these steps:

- 1.** Navigate to the following location: `yourmoodledirectory/theme/aardvark_pro/images/header`.
- 2.** In the header directory, you will see an image file entitled `logo.png`.
- 3.** Replace this file with the image that you want to appear at the top of your page. Make sure that you use the same, or similar, dimensions as the default image to ensure that your image is formatted correctly in the header area of the theme. Opening and editing the `logo.png` image is one way to accomplish this or, if you already have an image or logo that you wish to use, you may need to reduce its size to a height of not more than 100 pixels.
- 4.** The following screenshot is of a simple image created for this demonstration as it appears on the front page after being placed inside the header directory:



### ***What just happened?***

Through the very simple process of simply replacing one image with another that is given the same name, we modified the Aardvark theme header image. You can replace this image with another at any time using this process and only need to remember to make sure that the dimensions are appropriate to allow the image to be displayed properly.

## **Summary**

In this chapter, we have moved our Moodle site forward as a CIMS by first making small changes to three core file settings in order to modify the way categories are displayed, the way courses are displayed within categories, and to modify how many courses are displayed in a user's profile. These simple modifications enable us to better present a curriculum in a fashion that makes it easy to navigate and understand. We then installed the Content Pages block and set up several web pages inside our Moodle site with the block. Finally, we edited our third-party theme, Aardvark, in order to make the web pages we created easily accessible from the front page of our site. All of these tasks were performed with the purpose of adding an information 'skin' to our Moodle site that allows it to not only function as a VLE but also as an information portal for our educational program. In the next chapter, we will move inside the Moodle system and explore various methods for monitoring student access to the site and reporting on various activities for the purposes of communication, evaluation, and research.

