

# Programming Environment Overview

---

CPS 353: Internet Programming  
Professor Simon Miner  
[simon.miner@gordon.edu](mailto:simon.miner@gordon.edu)  
Last Revised: October 2013

## Introduction

This document provides you with details about the course programming environment in which you will code and submit your homework assignments and project milestones. Setup instructions for individual accounts are included with the descriptions of various software components of the environment below.

## Host and Operating System

The CPS353 programming environment resides on a virtual machine named **ips.cs.gordon.edu**, which is running Ubuntu Linux. Terminal access to this machine is available via SSH using a command like the following.

```
ssh -l username ips.cs.gordon.edu
```

Your user name is your last name in all lowercase letters. Your initial password is also your lowercased last name. You should change this as soon as possible with the *passwd* command.

```
passwd username
```

Your home directory is located at */home/username* and your shell is set to bash, giving you the ability to configure and customize your sessions on the ips box as you see fit.

## Web Server

An Apache 2 web server is installed and running on the ips box, allowing web access via the <http://ips.cs.gordon.edu> domain. You can host web content within the *public\_html* directory tree within your home directory. You'll need to create this folder and test its contents with the following commands.

```
cd ~  
mkdir public_html  
cd public_html  
echo "<h1>Hello, world!</h1>" > index.html
```

Now if you point your web browser to <http://ips.cs.gordon.edu/~username>, you should see a web page displaying the heading text "Hello, world!"

Your *public\_html* tree is capable of executing PHP scripts. You can test this functionality by running this command.

```
cd ~/public_html  
echo "<?php phpinfo(); ?>" > test.php
```

Then point your browser to <http://ips.cs.gordon.edu/~username/test.php> and you should get a page full of PHP configuration data.

## Homework

Each homework assignment should have its own dedicated directory within your *public\_html* tree. Homework directories should be named in the format *hw#*, where the “#” should be replaced with the number of the homework assignment in view (i.e. *hw1*, *hw2*, *hw3*, etc.). As much as possible, all files and resources pertaining to a given homework assignment should reside within that homework’s directory. (Do not reference files in one homework directory from another homework directory. Instead, make copies of the needed files in the new homework directory.)

Prior to the start of class on the date that each homework assignment is due, you must ensure that your completed work for the homework is present, accessible, and functioning correctly in that homework’s directory such that the professor can view and evaluate your completed work as follows:

- From the shell/command line in the `/home/username/public_html/hw#` directory.
- On the web at <http://ips.cs.gordon.edu/~username/hw#>.

Note that this is in addition to any materials that you are required to submit directly to the professor for a given homework assignment.

## Project Milestones

Just like homework assignments, each milestone for your major web development project must have its own dedicated directory beneath your *public\_html* tree root. Milestone directories must be named with the format *ms#* (i.e. *ms1*, *ms2*, *ms3*, etc.). Again, the content within each milestone directory should stand alone, not depending on resources from another milestone’s repository.

Prior to the start of class on the date that each project milestone is due, you must ensure that your completed work for the milestone is present, accessible, and functioning correctly in that milestone’s directory such that the professor can view and evaluate your completed work as follows:

- From the shell/command line in the `/home/username/public_html/ms#` directory.
- On the web at <http://ips.cs.gordon.edu/~username/ms#>

Note that this is in addition to any materials that you are required to submit directly to the professor for a given project milestone.

## Databases

A MySQL database server is installed and running on the ips machine. As the course progresses, you will need to administer and access objects on this server. Each student has two dedicated databases within the MySQL instance named *project\_username* and *testing\_username*. These serve the following purposes

- The project database is where your web development project’s main database will reside. The contents of this database will persist as you create the application.
- The tests database is intended for testing purposes. Its contents will be dropped and recreated each time you execute your application’s test suite.

You can log into the MySQL shell and access your databases via the following shell commands.

```
mysql -u username -p project_username  
mysql -u username -p testing_username
```

Your MySQL user has the same name as your operating system account on the ips machine (that is, your lowercased last name). Upon executing the above commands, you will be prompted for your MySQL user account's password. This starts out being the same as your MySQL username. You can and should change your password right away by executing the following command after logging into MySQL.

```
set password for 'username'@'localhost' = password('new-password');
```

## Ruby on Rails

Ruby and Ruby on Rails are installed on the server. There are a couple of places in your Rails project configuration and usage that you will need to specify custom values.

### Database Credentials

Your project's *database.yml* file includes numerous configuration values to control how Rails connects to your development, test, and production databases. (In this course, you will mostly use the development and test databases.) These values include the login credentials for each of your databases and include the following:

- *database* – the name of your project or testing MySQL database as defined above (e.g. *project\_username*, *testing\_username*).
- *username* – the name of your operating system and MySQL database user.
- *password* – the password to log into your database with your username. (Since this is a clear text value, you may want to use a throw-away value for this password.)

Note that whitespace is significant in YAML, so be sure to keep the (2 column) indentations, colons, and word-spacing intact when you change settings in this file.

### WEBrick Port Number

Ruby on Rails is bundled with a simple web server named WEBrick that is useful for developing your application. Since you and your classmates will be running separate instances of WEBrick on the ips.cs.gordon.edu machine (a single host), you will need to execute WEBrick on a pre-assigned port to avoid colliding with another student's copy of the server. By default, WEBrick listens on port 3000, but you can override this by including the *-p* flag in the command to launch the server:

```
rails server -p port
```

The instructor will provide you and your fellow students with distinct ports for WEBrick.

Note that you will sometimes find it useful to run the Rails server in the background as a daemon so that it stays running after you log out of your terminal session. This can be done by adding the *-d* switch to the "rails server" command.