

# Princeton University

## COS 333: Advanced Programming Techniques

### A COS 333 Computing Environment

*You should use your computer to do the COS 333 assignments. You must use the Princeton **courselab** cluster too. (COS 217 uses the similar but distinct **armlab** cluster.) The courselab cluster consists of two computers: **courselab03** and **courselab04**. When you log into courselab, you will be routed to either courselab03 or courselab04. The two computers share the same file system. Your courselab password is the same as your OIT password.*

*This document describes how to configure your courselab account and your own Mac, MS Windows, or Linux computer to do the COS 333 assignments.*

## 1. Configuring your courselab Account in General

*You may use courselab to develop your assignment solutions. You certainly will use courselab to test and submit your assignment solutions. So you must configure your courselab account such that it is appropriate for the course.*

(Mac and Linux) Open a terminal window, and issue the command `ssh yournetid@courselab.cs.princeton.edu` to log into courselab. Complete duo two-factor authentication.

(MS Windows) Using PuTTY, log into courselab (`courselab.cs.princeton.edu`). Complete duo two-factor authentication.

(All) Issue these commands on courselab:

```
$ # Copy configuration files to your home directory.
$ cp /u/cos333/.bashrc ~
$ cp /u/cos333/.bash_profile ~
$ cp /u/cos333/.emacs ~
```

**THE NEXT STEP IS NECESSARY! DON'T SKIP IT!**

(All) Log out of courselab by issuing an `exit` command, and log back in.

## 2. Configuring your courselab Account to Use Python

*You may use courselab to develop your assignment solutions. You certainly will use courselab to test and submit your assignment solutions. So you must create and configure on courselab an appropriate Python **virtual environment**. A Python virtual environment specifies a Python compiler/interpreter and the set of resources that it uses.*

(All) Issue these commands on courselab:

```
$ # Copy the pylint configuration file to your home directory.
$ cp /u/cos333/.pylintrc ~

$ # Create a directory to store your virtual environment(s).
```

```

$ mkdir ~/.virtualenvs

$ # Create a virtual environment named cos333.
$ python3 -m venv ~/.virtualenvs/cos333

$ # Activate your cos333 virtual environment.
$ source ~/.virtualenvs/cos333/bin/activate

$ # Upgrade pip to the most recent version.
$ python -m pip install --upgrade pip

$ # Install the coverage testing tool
$ # into your cos333 virtual environment.
$ python -m pip install coverage

$ # Install the pylint static code analysis tool
$ # into your cos333 virtual environment.
$ python -m pip install pylint

$ # Install the PyQt5 graphical user interface library
$ # into your cos333 virtual environment.
$ python -m pip install PyQt5

$ # Install the Flask web application framework
$ # into your cos333 virtual environment.
$ python -m pip install flask

$ # Deactivate your cos333 virtual environment.
$ deactivate

```

(All) To test: Issue these commands on courselab. In all cases the absence of error messages indicates success:

```

$ # Activate your cos333 virtual environment.
$ source ~/.virtualenvs/cos333/bin/activate

$ # Launch python and enter some statements.
$ python
>>> import coverage
>>> import pylint
>>> import PyQt5
>>> import flask
>>> quit()

$ # Deactivate your cos333 virtual environment.
$ deactivate

```

(All) Then throughout the semester issue these commands on courselab:

```

$ # Activate your cos333 virtual environment.
$ source ~/.virtualenvs/cos333/bin/activate

```

(Use Python as desired.)

```

$ # Deactivate your cos333 virtual environment.

```

```
$ deactivate
```

**(All) Shortcut:**

The `.bashrc` file contains this command:

```
alias activate333="source ~/.virtualenvs/cos333/bin/activate"
```

So you can issue the command `activate333` instead of the more verbose `source ~/.virtualenvs/cos333/bin/activate` to activate your `cos333` virtual environment.

### 3. Configuring Your Computer to Use X Windows

*You will run graphical programs (for example, firefox) on courselab, and have those graphical programs display their windows on your computer. So you must install an X Windows server on your computer.*

(Mac) Follow these instructions to install the Xquartz X Windows server:

Browse to <http://xquartz.macosforge.org/landing/>

Click on the `Xquartz-2.8.1.dmg` link to begin a download.

After the download completes, in the Finder double click on the `XQuartz-2.8.1.dmg` file to perform the installation.

(Linux) An X Windows server already is installed on Linux systems.

(MS Windows) Follow these instructions to install the VcXsrv X Windows server:

Browse to <http://sourceforge.net/projects/vcxsrv>

Click on the `Download` button to begin a download.

After the download competes, in Windows Explorer double click on the `vcxsrv-64.1.20.9.0.installer.exe` file to perform the installation. Use the default install settings.

(Mac and Linux) To test:

Login into courselab using the command `ssh -Y yournetid@courselab.cs.princeton.edu`. Note the `-Y` option.

On courselab issue the command `xclock &`. A window displaying a clock should appear on your computer.

(MS Windows) To test:

Launch VcXsrv on your local computer. Use the default settings.

Launch PuTTY.

Change PuTTY settings as usual (Window → Colours → Use system colours). Then change one more setting: make sure the Connection → SSH → X11 → X11 Forwarding → Enable X11 forwarding check box is checked.

Use PuTTY to log into `courselab.cs.princeton.edu`.

On courselab issue the command `xclock &`. A window displaying a clock should appear on your computer.

(Mac and Linux) Thereafter, to use courselab to run graphical applications... Use the Terminal application to login to courselab. However, specify the `-Y` option to ssh:

```
ssh -Y yournetid@courselab.cs.princeton.edu
```

(MS Windows) Thereafter, to use courselab to run graphical applications:

Launch VcXsrv on your local computer. Use the default settings.

Launch PuTTY.

Change PuTTY settings as usual (Window → Colours → Use system colours). Then change one more setting: make sure the Connection → SSH → X11 → X11 Forwarding → Enable X11 forwarding check box is checked.

Use PuTTY to log into `courselab.cs.princeton.edu`.

## 4. Selecting the Login Shell for Your Computer

*If your computer is a Mac or Linux system, then you will use a **shell**. You should use the **bash** shell.*

*Your **login shell** is the shell that executes automatically when you open a terminal window. On most Linux distributions, the default login shell is **bash**. On Mac systems that run operating systems older than macOS 10.15 Catalina, the default login shell also is **bash**. However, on Mac systems that run macOS 10.15 Catalina or newer, the default login shell is **zsh**.*

*You should make sure that your default login shell is **bash**.*

(Mac and Linux) From a terminal, issue the command `printenv SHELL`. If the output is `/bin/bash`, then your login shell is **bash**. If the output is not `/bin/bash`, then issue the command `chsh -s /bin/bash`, enter your password when prompted, exit your terminal session, start a new terminal session, and again issue the command `printenv SHELL` to confirm that the output is `/bin/bash`.

## 5. Configuring Your Computer to Use Python

*You should use your computer to develop your assignment solutions. So you should install **Python 3.8** on your computer. (Python 3.6, 3.7, or 3.9 probably will work, but there are no guarantees.). Then you should create and configure on your computer an appropriate Python virtual environment..*

(All) Create a file in your home directory named `.pylintrc` whose contents are the same as those of the courselab `/u/cos333/.pylintrc` file.

(Mac) Browse to <https://www.python.org/downloads/>. Click on the *Download* link for the newest version of Python 3.8. In the resulting page click on the *macOS 64-bit installer* link. In the Mac Finder, double click on the resulting .pkg file to install. Follow the instructions at the end of the installation process to install certificates. To test, issue the command `python3.8` at a shell prompt in a Terminal window. The Python interpreter/compiler should launch, and identify itself as Python 3.8. To exit, at the `>>>` prompt enter the `quit()` statement.

(MS Windows) Browse to <https://www.python.org/downloads/>. Click on the *Download* link for the newest version of Python 3.8. In the resulting page click on the *Windows x86-64 executable installer* link. In Windows Explorer, double click on the resulting .exe file to install. During the installation, check the appropriate checkbox to indicate that you want the installer to add python to your path. To test, issue the command `python` in a Command Prompt window. The Python interpreter/compiler should launch, and identify itself as Python 3.8. To exit, at the `>>>` prompt enter the `quit()` statement.

(Linux) Use your Linux package manager to download and install the `python3.8` and `python3.8-venv` (or similarly named) packages. To test, issue the command `python3.8` at a shell prompt in a terminal window. The Python interpreter/compiler should launch, and identify itself as Python 3.8. To exit, at the `>>>` prompt enter the `quit()` statement.

(Mac and Linux) Issue these commands in a terminal window:

```
$ # Create a directory to store your virtual environment(s).
$ mkdir ~/.virtualenvs

$ # Create a virtual environment named cos333.
$ python3.8 -m venv ~/.virtualenvs/cos333

$ # Activate your cos333 virtual environment.
$ source ~/.virtualenvs/cos333/bin/activate

$ # Upgrade pip to the most recent version.
$ python -m pip install --upgrade pip

$ # Install the coverage testing tool
$ # into your cos333 virtual environment.
$ python -m pip install coverage

$ # Install the pylint static code analysis tool
$ # into your cos333 virtual environment.
$ python -m pip install pylint

$ # Install the PyQt5 graphical user interface library
$ # into your cos333 virtual environment.
$ python -m pip install PyQt5

$ # Install the Flask web application framework
$ # into your cos333 virtual environment.
$ python -m pip install flask

$ # Deactivate your cos333 virtual environment.
$ deactivate
```

(MS Windows) Issue these commands in a Command Prompt window:

```

C:\>REM Create a directory to store your virtual environment(s).
C:\>mkdir %HOMEPATH%\virtualenvs

C:\>REM Create a virtual environment named cos333.
C:\>python -m venv %HOMEPATH%\virtualenvs\cos333

C:\>REM Activate your cos333 virtual environment.
C:\>%HOMEPATH%\virtualenvs\cos333\Scripts\activate.bat

C:\>REM Upgrade pip to the most recent version.
C:\>python -m pip install --upgrade pip

C:\>REM Install the coverage tool.
C:\>REM into your cos333 virtual environment.
C:\>python -m pip install coverage

C:\>REM Install the pylint static code analysis tool.
C:\>REM into your cos333 virtual environment.
C:\>python -m pip install pylint

C:\>REM Install the PyQt5 graphical user interface library.
C:\>REM into your cos333 virtual environment.
C:\>python -m pip install PyQt5

C:\>REM Install the Flask web application framework.
C:\>REM into your cos333 virtual environment.
C:\>python -m pip install flask

C:\>REM Deactivate your cos333 virtual environment.
C:\>deactivate

```

(Mac and Linux) To test, issue these commands in a terminal window. In all cases the absence of error messages indicates success:

```

$ # Activate your cos333 virtual environment.
$ source ~/.virtualenvs/cos333/bin/activate

$ # Launch python and enter a few statements.
$ python
>>> import coverage
>>> import pylint
>>> import PyQt5
>>> import flask
>>> quit()

$ # Deactivate your cos333 virtual environment.
$ deactivate

```

(MS Windows) To test, issue these commands in a Command Prompt window. In all cases the absence of error messages indicates success:

```

C:\>REM Activate your cos333 virtual environment.
C:\>%HOMEPATH%\virtualenvs\cos333\Scripts\activate.bat

```

```
C:\>REM Launch python and enter a few of statements.
C:\>python
>>> import coverage
>>> import pylint
>>> import PyQt5
>>> import flask
>>> quit()

C:\>REM Deactivate your cos333 virtual environment.
C:\>deactivate
```

(Mac and Linux) Then throughout the semester issue these commands in a terminal window:

```
$ # Activate your cos333 virtual environment.
$ source ~/.virtualenvs/cos333/bin/activate
```

(Use Python as desired.)

```
$ # Deactivate your cos333 virtual environment.
$ deactivate
```

(MS Windows) Then throughout the semester issue these commands in a terminal window:

```
C:\>REM Activate your cos333 virtual environment.
C:\>%HOMEPATH%\virtualenvs\cos333\Scripts\activate.bat
```

(Use Python as desired.)

```
C:\>REM Deactivate your cos333 virtual environment.
C:\>deactivate
```

#### **(Mac and Linux) Shortcut suggestion:**

Make sure you have a `.bash_profile` file in your HOME directory, and that it contains this command:

```
source .bashrc
```

Make sure you have a `.bashrc` file in your HOME directory, and add this command to it:

```
alias activate333="source ~/.virtualenvs/cos333/bin/activate"
```

Then you can issue the command `activate333` instead of the more verbose `source ~/.virtualenvs/cos333/bin/activate` to activate your cos333 virtual environment.

#### **(MS Windows) Shortcut suggestion:**

Create a directory named `bin` immediately subordinate to your home directory. That is, create a directory named `%HOMEPATH%\bin`. In that directory create a file named `activate333.bat` that contains this line:

```
%HOMEPATH%\virtualenvs\cos333\Scripts\activate.bat
```

Then in *Control Panel* navigate to *System and Security* → *System* → *Advanced System Settings* → *Environment Variables*. In the *User Variables for yourloginid* area:

If the Path environment variable exists, then add the directory `%HOMEPATH%\bin` to its value.

If the Path environment variable does not exist, then create it such that its value is the directory %HOMEPATH%\bin.

Having performed those steps you can issue the command `activate333` instead of the more verbose `%HOMEPATH%\virtualenvs\cos333\Scripts\activate.bat` to activate your cos333 virtual environment.

## 6. Configuring Your Computer to Use the SQLite Client

*You should use the SQLite client – a program whose name is `sqlite3` – to access SQLite databases that reside on your computer. So you should install the `sqlite3` program.*

(Mac) The `sqlite3` program probably is bundled with your operating system. To test, in a terminal window issue the command `sqlite3`. Make sure `sqlite3` launches. Enter the command `.quit` to exit the program.

(Mac) If the `sqlite3` program is not bundled with your operating system, then browse to <https://www.sqlite.org/download.html>, and download the file named `sqlite-tools-osx-x86-3330000.zip`. In the Finder double click on the downloaded file to unzip it, thereby creating a directory that contains the `sqlite3` file. To test, in a terminal window issue the command `path/sqlite3`, where `path` is the directory that contains the `sqlite3` file. Make sure `sqlite3` launches. Enter the command `.quit` to exit the program.

(MS Windows) Browse to <https://www.sqlite.org/download.html>, and download the file named `sqlite-tools-win32-x86-3340000.zip`. In Windows Explorer unzip the downloaded file, thereby creating a directory that contains the `sqlite3.exe` file. To test, in a Command Prompt window issue the command `path\sqlite3.exe`, where `path` is the directory that contains the `sqlite3.exe` file. Make sure `sqlite3` launches. Enter the command `.quit` to exit the program.

(Linux) Use your Linux package manager to download and install the `sqlite3` (or some similarly named) package. To test, in a terminal window issue the command `sqlite3`. Make sure `sqlite3` launches. Enter the command `.quit` to exit the program.

## 7. Copying Files Between Your Computer and Your Teammate's Computer

*Throughout the semester you will need to copy assignment files between your computer's file system and the file system of your teammate's computer.*

*A simple approach is to use a **Google drive**. You can create a private Google drive, and share it with your teammate. You can copy files from your computer's file system to your Google drive, and your assignment teammate can copy those files from your Google drive to his/her computer's file system. And vice versa.*

*In many cases you will find it more convenient to use **GitHub**. For each assignment you can create a private Git repository on GitHub, and share it with your teammate. You can clone the repository to your computer, and your teammate can clone the repository to his/her computer. You then can push files from your computer's repository to the GitHub repository, and your teammate can pull files from the GitHub repository to his/her computer's repository. And vice versa.*



## 8. Copying Files Between Your Computer and courselab

*Throughout the semester you (or your assignment teammate) will need to copy assignment files between your computer's file system and the courselab file system.*

*A simple approach is to use **FileZilla**. That is, you can use the FileZilla application to upload files from your computer's file system to the courselab file system, and download files from the courselab file system to your computer's file system. FileZilla uses SFTP, the secure file transfer protocol.*

*You may have used FileZilla in COS 217. However, now that the Computer Science Department has adopted duo two-factor authentication, using FileZilla is trickier than it used to be.*

One time only...

(Mac and MS Windows) Download the FileZilla client from <http://filezilla-project.org/> and install it.

(Linux) Use your Linux package manager to download and install the filezilla (or some similarly named) package.

(All) Launch FileZilla.

(All) Create a new Site.

From the menu choose *File* → *Site Manager...*

Click the *New Site* button.

Give the new site the name `courselab`.

In the *General* tab

Set the *Protocol* to SFTP - SSH File Transfer Protocol

Set the *Host* to `courselab.cs.princeton.edu`

Set the *Port* to 22

Set the *Logon Type* to Interactive

Set the *User* to your login id

In the *Transfer Settings* tab:

Check the *Limit number of simultaneous connections* check button

Set the *Maximum number of connections* to 1

(If you omit those settings, then FileZilla will require you to enter your password and respond to a telephone call or text message each time you transfer a file. For example, if you were to select 5 files in the left-side pane and drag them to the right pane, then you would need to enter your password 5 times and respond to 5 telephone calls or text messages!)

Click the *OK* button.

(All) Exit FileZilla.

Repeatedly throughout the semester as desired...

(All) Launch FileZilla.

(All) Connect to courselab.

From the menu choose *File* → *Site Manager...*

Select *courselab*.

Click the *Connect* button.

In the resulting dialog box enter your password and click the *OK* button.

In the resulting dialog box, enter an integer to indicate the kind of callback that you desire. (For me 1 indicates a telephone call to my cell phone, 2 indicates a telephone call to my landline phone, and 3 indicates SMS passcodes.) Click the *OK* button.

(All) Transfer files as desired.

Repeatedly click in the left-side pane to navigate through your computer's file system.

Repeatedly click in the right-side pane to navigate through the courselab file system.

Drag-and-drop the name of the desired file from the right pane to the left pane, thus copying the file from the courselab file system to your computer's file system. Or drag-and-drop the name of the desired file from the left pane to the right pane, thus copying the file from your computer's file system to the courselab file system.

*In many cases you will find it more convenient to use **GitHub**. For each assignment you can create a private Git repository on GitHub. You can clone the repository to your computer, and clone the repository to courselab. You then can push files from your computer's repository to the GitHub repository, and pull files from the GitHub repository to your courselab repository. And vice versa. Your teammate can use the same approach.*

Copyright © 2022 by Robert M. Dondero, Jr.