

Princeton University

COS 217: Introduction to Programming Systems

A Minimal COS 217 Computing Environment

1. Access the COS 217 Account on Ed

- 1.1. You can access Ed through Canvas.
- 1.2. Post questions and comments (that comply with the course communication policies) to Ed. Posts will be available to all other students and instructors. Remember to check Ed often, especially while working on assignments and preparing for exams.

If you enroll in the COS 217 course after the start of the semester, then there will be a delay – typically a day or two – before you have authorization to perform the following steps.

2. Conducting an armlab Terminal Session

- The COS 217 course uses the armlab computer cluster. The cluster consists of 2 computers named armlab05 and armlab06. Both use the same file system.
- The first time you log into an armlab computer, you will be using a bare-bones computing environment. Only after you configure the Bash shell (see the *Configuring the Bash Shell* section of this document) will your environment match what is expected to complete your COS 217 work.
- Your computer communicates with an armlab computer via any terminal application that can use the SSH (secure shell) protocol. Three such programs are GitBash (for MS Windows) and Terminal (for Mac OS X), and PuTTY.

Repeatedly throughout the semester as required...

2.1. Option 1: Use a computer running Microsoft Windows.

If you purchased your computer through Princeton or have installed the infrastructure for COS 126 or COS 226, you already have the GitBash terminal emulator available on your computer and do not need to install a different one.

Many IDEs (integrated development environments) have their own built-in terminal emulators available from within the application.

If your computer is running Microsoft Windows and you did not purchase your computer through Princeton in recent years and did not take COS 126 or COS 226 and you are not using an IDE, then you may need to download and install a terminal emulator that can use SSH. Here are instructions for two such applications:

GitBash (more recommended): <https://www.atlassian.com/git/tutorials/git-bash>

PuTTY (less recommended): <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Launch Your Terminal Emulator Software.

Using Windows Explorer, double-click on the exe file for GitBash, PuTTY, or another applicable program.

Log into the armlab computer.

In GitBash or another terminal emulator:

Enter this command, replacing *YOURNETID* with your Princeton netid:

```
ssh YOURNETID@armlab.cs.princeton.edu
```

Enter your Princeton password. (The password will not echo as you type.)

Confirm that the window displays a Linux shell prompt.

Now you can use the armlab computer via your terminal emulator as desired.

Log out of the armlab computer when you have finished your work session.

In your terminal emulator window, issue the `exit` command to disconnect from the armlab computer. Depending on which software you are using, the window may or may not close automatically.

2.2. Option 2: Use a computer running Mac OS X.

Open a Terminal window.

Launch Spotlight by clicking on the magnifying glass icon at the right side of the menu bar. Type *Terminal* in Spotlight. In the resulting pop-up list click on *Terminal*. (For most Mac setups, ⌘-spacebar will also launch Spotlight.)

Log into the armlab computer.

In the Terminal window:

Enter the command:

```
ssh YOURNETID@armlab.cs.princeton.edu
```

where *YOURNETID* is your Princeton netid.

If an SSH-related message appears, enter `yes`.

Enter your Princeton password. (The password will not echo as you type.)

Use the armlab computer via the Terminal window as desired.

Log out of the armlab computer when you have finished your work session.

In the Terminal window, enter the `exit` command.

Close the Terminal window.

Click on the red button at the upper left of the Terminal window.

3. Configuring the Bash Shell

One time only, in an armlab terminal session...

3.1. Enter the command `printenv SHELL` and confirm that the output is `/bin/bash`.

3.2. Enter these two commands to copy reasonable Bash configuration files to your home directory:

```
cp /u/cos217/.bash_profile /u/YOURNETID
```

where *YOURNETID* is your Princeton netid.

Reply to the `cp: overwrite '.bash_profile'?` question by entering `y`.

```
cp /u/cos217/.bashrc /u/YOURNETID
```

where *YOURNETID* is your Princeton netid.

Reply to the `cp: overwrite '.bashrc'?` question by entering `y`.

- 3.3. Suggestion: Enter the `cat .bashrc` and `cat .bash_profile` commands to examine the contents of the `.bashrc` and `.bash_profile` files.

The changes take effect during your **next** armlab terminal session.

4. Configuring the Emacs Editor

One time only, in an armlab terminal session...

- 4.1. Enter this command to copy a reasonable Emacs configuration file to your home directory:

```
cp /u/cos217/.emacs /u/YOURNETID
      where YOURNETID is your Princeton netid.
```

- 4.2. Suggestion: Enter the `cat .emacs` command to examine the contents of the `.emacs` file.

5. Configuring the Splint Source Code Checker

One time only, in an armlab terminal session...

- 5.1. Enter this command to copy a reasonable Splint configuration file to your home directory:

```
cp /u/cos217/.splintrc /u/YOURNETID
      where YOURNETID is your Princeton netid.
```

- 5.2. Suggestion: Enter the `cat .splintrc` command to examine the contents of the `.splintrc` file.

6. Copying Files between armlab and Your Computer

You can use `github` to create a repository of each assignment. Details and other options will be covered in lectures and/or A0.

Copyright © 2026 by Robert M. Dondero, Jr., Xiaoyan Li, Donna Gabai, Christopher Moretti