

# Helpful Information for Programming Assignments

CSc 345 – Summer 2014

Instructor: Qiyam Tung

I have been gathering content that some students may not be familiar with. Here is my list to accommodate those gaps. To be clear, I do not think that much of this needs to be covered in this course (because it should have been covered). However, because I don't have perfect information, I'm putting all that information here.

## Working on lectura

I believe you all should know how to submit files to and from `lectura` by now. However, you may be less familiar with doing work (i.e. editing files, running programs) on it.

## Logging on

To log on to `lectura`, you will need a program that gives you remote access. For Windows users, you can use the program PuTTY (Google it). Once you have it, in the “Host Name” text field, type in `lectura.cs.arizona.edu`. It will then prompt you for your username and password.

If you are using a Mac or Linux OS, then open up the terminal window and type in

```
ssh username@lectura.cs.arizona.edu
```

It will then prompt you for your password.

## Commands

Here is a list of useful commands

- `cd`: changes directories. Examples:
  - `cd ..` takes you to the parent directory.
  - `cd ~` takes you to your home directory. That is, the directory with your personal files (as opposed to system files/programs which everyone uses).
  - `cd cs345` takes you to the `cs345` directory, but only if it exists.
- `ls`: lists the files and directories in the directory you're currently in.
  - `ls` lists the files in the current directory.
  - `ls -l myprog.java` lists the information about your program `myprog.java` (includes size, date of modification, etc.)
- `pwd`: prints what current directory you're in.
- `javac`: compiles a Java source file.
  - `javac cs345_prog1.java` compiles the single Java source file. Note that if `cs345_prog1.java` depends on other classes, then you need to compile those as well.
  - `javac *` compiles all the Java source files in the current directory.
- `java`: runs the Java virtual machine (i.e. the command to run your program).
  - `java cs345_prog1` runs the Java program `cs345_prog1`. Note the absence of the `.java` when calling the program.
  - `java cs345_prog1 argument1 argument2` runs the program with the arguments `argument1` and `argument2`.

Here is an example of some commands I ran on lectura after logging in. Basically, I am finding my way back to my directory with my program and compiling and running it.

```
Last login: Wed Jul 2 13:37:54 2014 from sullivan.cs.arizona.edu
qtung@lectura~$ pwd
/home/qtung
qtung@lectura~$ ls
7zip                home                Templates
backup             libpeerconnection.log  test
bin                login.generic         test_back_dir
classes            mail                 test_back_dir_1
C:\nppdf32Log\debuglog.txt Mail                 test_back_dir_2
color_theme        mailrc               test.bash
data               matlab               test_dir_1
dead.letter        misc                 test_dir_2
Desktop            Music                 test_dir_3
Documents           News                 test_dir_4
Documents and Settings personal            test_dir_5
DOTbashrc          Pictures              tmp
DOTprofile         Public                Ubuntu One
Download           python_home           Videos
Downloads          python_libs           VirtualBox VMS
Dropbox            research              vision
EclipseWorkspace  runtime-EclipseApplication winrar
euler              save                  winscp.RND
ffmpeg_libs        scripts               workspace
ffpresets          software              xinitrc
file1.txt          synergy.conf          zealot.PNG
file2.txt          ted_video

qtung@lectura~$ cd classes
qtung@lectura~$ ls
backup  cs433  cs460  cs525.zip  cs577  mat564  math571a
cs345  cs433.zip  cs473  cs538  cs630  mat566  teaching
cs352  cs445  cs520.zip  cs545  cs645  math513

qtung@lectura~$ cd cs345
qtung@lectura~$ ls
course_outline.txt  licence          spring09  summer2006  webpage
fall10              #missing_grades#  spring10  summer2010
fall12              others           spring11  summer2011
JAVA3e20110520.pdf  program1         summer14  summer_2012

qtung@lectura~$ cd summer14
qtung@lectura~$ ls
cs345_summer_12_files  index.html~      slide_match.txt  syllabus.html~
demo                  lectura_submission.pdf  slides           ta_info
exams                 lectures          space.2014-07-01  tmp.txt
files                 moon_material     submissions
homework              participations    syllabus_files
index.html            programs          syllabus.html
```

```

qtung@lectura~$ ls -l lectura_submission.pdf
-rw-rw-r-- 1 qtung qtung 26269 Jul  2 14:57 lectura_submission.pdf
qtung@lectura~$ cd programs/
genetic_algorithm/ sorting/
qtung@lectura~$ cd programs/sorting/
qtung@lectura~$ ls
amoeba.jpg                helpful_info.tex~
amoeba_sorted.png        ImageLibrary.class
Aurora-Blancanieves_sorted.png  ImageLibrary.java
aurora.jpg                sample_files
aurora_sorted.png        sorting.aux
ComparableImage.class    sorting.log
ComparableImage.java     sorting.out
cs345_prog1.class        sorting.pdf
cs345_prog1$ComparableImage.class  sorting.tex
cs345_prog1.java         sorting.tex~
helpful_info.aux         test.png
helpful_info.log        tiger.jpg
helpful_info.out        tiger_not_divisible_by_10.jpg
helpful_info.pdf        tiger_sorted.png
helpful_info.tex
qtung@lectura~$ javac *.java
qtung@lectura~$ java cs345_prog1 aurora.jpg
Sorting aurora.jpg as aurora_sorted.png
  Num writes: 6428650
qtung@lectura~$ ls -l aurora_sorted.png
-rw-rw---- 1 qtung qtung 534612 Jul  2 16:09 aurora_sorted.png
qtung@lectura~$

```

## Program Arguments

### What is a Program Argument?

For those of you who have only done programming in 227/127B, you may not be aware what a program argument is.

Very briefly, when you run a program, it can accept arguments that can change its behavior, much like how a Java method can accept arguments.

In your Java program, they are the `args` parameter that is always there in your `main` method. The following program will print all the program arguments.

```

public static void main(String[] args)
{
    for (int i = 0; i < args.length; i++)
        System.out.println(args[i]);
}

```

Running your program will give the following output

```
bash$ java cs345_prog1 hello world
hello
world
```

## Running a Program with Arguments

It is straightforward how to do this on the shell (as seen in the homework 4 document). If you are used to programming in Eclipse, you may not know how to run it. Take a look at this tutorial <http://www.cs.colostate.edu/helpdocs/eclipseCommLineArgs.html>.