TOOLS FOR CS200

RECITATION 1

If you can't log into the computers, let me know.

Submitting Homeworks

1. Open a terminal and navigate to any file you want to submit (I'll call it `myFile`)
2. Type **`~cs200/bin/checkin HW0 myFile`** to submit myFile as Homework 0.
3. Most homework assignments will require their files submitted in the form of a **`.tar`** archive. ***This archive must contain only the source code of the assignment.*** No folders, **`.class`** files, or anything else. If this rule is not followed, 5 points will be deducted.
4. There are two ways to make a **`.tar`** archive for Homework number X:
   (a) Open a terminal in the folder containing your source code, do **`tar -cf HWX.tar *.java`**
      i. This will make the file **`HWX.tar`**, which will contain all the **`.java`** files in the terminal's current directory.
      ii. *Be careful with this command*, backwards arguments can destroy data!
   (b) Using the file browser (Activities -> Files, on the left) which is a GUI file browser, you can right-click on any folder and select **`Compress...`** In the next dialog name the file appropriately and select .tar as the type. Then click Create.
5. To check in Homework number X as file HWX.tar, type **`~cs200/bin/checkin HWX HWX.tar`**
6. In general, checkin commands have the following format: **`~cs200/bin/checkin <name> <file>`**
   (a) where **`<name>`** is the name of the Homework / Recitation / whatever, and **`<file>`** is the name of the file you want to submit
7. Be sure to read the output generated by checkin, if there were any problems checking something in it will tell you about them.

Submitting Recitations

1. Sign the attendance sheet. You will not receive credit if you do not sign it.
2. Create a new Java project and follow the instructions given for the particular recitation you are in. For today, make a Hello World program.
3. If you are enrolled in Lab 0Y and are submitting Recitation X, the name is RXL0Y and the file name should be RXL0Y.tar, so the checkin command is: **`~cs200/bin/checkin RXL0Y RXL0Y.tar`**
   (a) Recitation Numbering: Monday is **`L01`**, Tuesday is **`L02`**, Wednesday is **`L03`**, Thursday is **`L04`**.
4. Recitations will primarily be graded on attendance.

Tips and Tricks

1. Eclipse stores source code in the *Workspace* by default. Most people's workspace is located at **`~/workspace`**, a.k.a in their Home folder.

    (a) Each project gets a directory with in the workspace containing bin and src folders. The src folder contains the project's source code. So source code for a project called PA1 would be in the folder **`~/workspace/PA1/src`**

    (b) In Eclipse, select a file in the Package Explorer and then go to File -> Properties (or hit Alt-Enter), then select Resource on the left. The Location field tells you where that file is stored.

2. **`~/`** is a terminal short cut for your home folder.

3. If you are in the system's file browser (Activities -> Files), right click on empty white space and select **`Open In Terminal`**. This will open a terminal window which has already been navigated to the folder your file browser was in. This can save a lot of time.

4. My language is a bit sloppy, folder and directory mean the same thing.

5. Helpful terminal commands:

    (a) **`man`** - get help with terminal commands (`man ls` will get help with the ls command, type q to quit man)

    (b) **`ls`** - list contents of current directory

    (c) **`cd`** - change directory (use **`cd ..`** to go back a directory)

    (d) **`pwd`** - print out which directory the terminal is in

    (e) **`cp`** - copy a file

    (f) **`mv`** - move or rename a file

    (g) **`ssh`** - log into a remote computer and use it's terminal. Remote machine must be running a ssh server, most Linux machines run one by default. There are Windows versions of ssh available, I use one called PuTTY.

    (h) **`scp`** - copy a file to / from a remote computer. Essentially it is **`cp`** via **`ssh`**. Windows versions are available.

    (i) There are MANY more than this.

6. Java can be easily compiled and run outside of Eclipse, from the terminal.

    (a) **`javac`** is the compiler command. **`javac helloworld.java`** compiles the file helloworld.java. **`javac *.java`** compiles all `.java` files in the current directory.

    (b) compiling a `.java` file results in a `.class` file

    (c) **`java`** is the command used to invoke Java. Run java on the `.class` file which contains the main method of your program (leave off the .class extension). If my main method was in `helloworld.java`, which was compiled to `helloworld.class`, then **`java helloworld`** would run the program.

7. There has been an error where you are unable to create a new java class in eclipse, the 'finish' button in the class creation dialog does not work. The solution is to revert all of eclipse's setting to their defaults. *This will undo any custom settings that you have.*

    (a) Close eclipse, and run in a terminal:

    ```
    cd;
    mv .eclipse .eclipse_backup;
    eclipse.sh;
    ```

    (b) If the problem is fixed, and you're happy with it, you can delete .eclipse_backup