Lab stuff available via http://www.cs.albany.edu/~sdc/CSI201 too.

Think about visiting or even switching to a Friday lab: They are much less crowded! Today you need your UAlbany NetId and password again. If you don't have them but have your UAlbany PIN, just reset your password on MyUAlbany. Otherwise, bring photo id to LC-27. Get them right away and, if necessary, complete this lab in one of the other meetings this week.

1. (Unless you're logged in already) drag the mouse to or below the bottom of the screen near the middle, and click the boxy black (SSH to ITSUNIX) icon. It's leftmost in the row of icons.

At the login prompt, log in with your NetID (your initials and some numbers) and your MyUAlbany password. You will NOT SEE any of your password echoed back when you type it! That's intended to prevent other people from learning your (secret) password by looking over your shoulder.

You will see something like: (If not, try correcting any mistakes in your NetID and password.)

Sun Microsystems Inc.  SunOS 5.10  Generic January 2005
********************************************************************
..... many lines omitted ...
unix2%
What you see is the Unix shell prompt. (The 2 might be different). Do nothing, but go on to Step 2.

2. Below, use the keyboard driven shell command line to verify you had set up proper folders/directories for CSI201 material, including the Book Classes. If you find you did not, you must switch to the other lab directions to get that set up right away.

(You will do step 2 just once during the whole semester, if it goes well. If not, GET HELP to get past the technical glitches as soon as possible!)

Type the following Linux/Unix shell commands, exactly as shown, and press the enter key to make the shell perform each one. Notice all the letters are lower case.

```
cd
pwd
```

You should see something like

```
/home1/s/m/sml23456
unix2%
```

The letters, for example s and m here, and the numbers will be from your NetId. The cd command made the shell reset its working directory to the default. (That default is called your "home" directory.) The pwd command makes the shell print the path name of the working directory, which is the directory (or folder) where you are currently working.

Now type: `cd CSI201` and press ENTER. If you see ANYTHING, like an error message, IN ADDITION TO the prompt unix2%, GET HELP right away or switch to the other lab directions.

Finally, type `ls -l`

That is two lower-case letter l-s, NOT ones! (ell) s (ess) space hyphen (ell)

You should see the uncompressed book classes folder/directory that was supposed to result from your doing Homework01. If not, GET HELP or switch to the other lab directions.

You should see something like

```
total 7344
drwx----- 2 sdc faculty 512 Jan 4 16:33 CSI201
unix2%
```
3. Copy an entire directory/folder named **Lab01** from our common disk area into and under your CSI201 directory/folder,
   The shell command for that is:
   \[ \text{cp -r /usr/local/depts/cs/geintro/Lab01 .} \]
   There is a little dot (period) and a space before the dot at the end of the line. You must type them both! \texttt{cp} means copy; hyphen-\texttt{r} is the option to do it recursively, meaning copy everything below \texttt{Lab01} also.
   
   What should happen is you get a new folder/directory named \texttt{Lab01} under \texttt{CSI201}. You will find in it two .\texttt{java} files (when you command \texttt{ls -l CSI201} to the shell.):
   \begin{itemize}
   \item \texttt{DrawWithMethodsApp.java}
   \item \texttt{ArtisticTurtle.java}
   \end{itemize}

4. Type in EXACTLY what you see below, which is a line of lower-case letters and slash characters with no spaces in-between. It is our UNIX shell command to start DrJava in the lab, so you must press ENTER at the end of the line:
   \[ /usr/local/depts/cs/geintro/drjava \]
   After a few moments the DrJava splash screen will appear, followed by the DrJava window itself. If it offers an upgrade, click "Not Now" or "No".

5. (AGAIN) set the "Extra Classpath" preference/resource value to locate your copy of the BookClasses and (Get bookclasses if necessary: ASK FOR HELP!)

6. Exit and restart DrJava. This time, you might want to make the window much bigger.

7. Starting from the Open or File/Open menus, navigate up and down to get to your 
   \[ /home/stuff...stuff/netID/CSI201/Lab01 \] directory/folder (get as much help as you need!), and open \texttt{ArtisticTurtle.java} and then \texttt{DrawWithMethodsApp.java}

8. Verify that our \texttt{DrawWithMethodsApp} application compiles and runs. **DON'T GO ON IF IT DOESN'T MAKE THE DRAWING BELOW.** (Words "Red" and "Green" will NOT be drawn.)

9. Inside the \texttt{main(...)} method definition in \texttt{DrawWithMethodsApp}, disable or delete the \texttt{tref.forward( ... )} and \texttt{tref.turn( ... )} method calls so all of the actual drawing and turning instructions are instructed from within the \texttt{ArtisticTurtle}'s code, not from inside \texttt{main(...)}. It's sometimes a good idea to disable a Java line but still leave it written in the program file by making the instruction into a comment: Just precede it with //

10. Save, compile and run to verify the effects of your edits to \texttt{DrawWithMethodsApp.java} Your \texttt{DrawWithMethodsApp} should make the artwork below: *(GET HELP IF NOT!!)*
Inside the hook( param ) method definition in ArtisticTurtle, YOU finish our code to draw a two-lined hook. Make sure it leaves the ArtisticTurtle pointing UP.

Your first job is to study the hook(...) method's instructions inside hook's body. They tell this ArtisticTurtle to draw ONLY ONE LINE, and then to turn right 90 degrees ONLY once.

That is not the way to draw one whole hook, because a hook is made of TWO lines, not one. After drawing the first line (length 50 pixels) and turning right, it's not finished. You have to type in the Java code to tell it to draw the second line (length 35 pixels, shorter). Which Turtle should go forward? this

Also, after drawing the second line horizontally, this ArtisticTurtle must be told to turn -90 degrees (which is 90 degrees LEFT) so he/she/it finishes pointing upward. (You) please type in another Java instruction line to command this Turtle to do that turn operation.

11. Save, compile and run to verify the effects of your edits to DrawWithMethodsApp.java Your DrawWithMethodsApp should now make the artwork below(except for the words): (GET HELP IF NOT!!)

Three step stairway to Java glory.

All Red

All Red

12. After and outside the lab, you must retrieve your edited ArtisticTurtle.java file AND your DrawWithMethodsApp.java file from the UAlbany "S:" drive and upload both of them, plus the two ArtisticTurtle.class and DrawWithMethodsApp.class files, to prove to your TA that you got your modifications to compile and so we can test them.

You are expected to use some computer (Linux, MAC, Windows) outside of lab to do the programing projects to be assigned during the semester. That computer could (preferably) be your own laptop, net-book or desktop, or one in the Information Commons rooms in each UAlbany Library. You will have to do steps 2, 5, 6 and maybe others to set up that computer just once.

BUT: you will repeat steps like 1, sometimes 3, 4, and steps at and after 7 over and over again throughout the semester.

If you were to now leave the lab, and go on one of the Information Commons computers in the UA libraries, you will find your CSI201 directory/folder, the Lab01 subdirectory and their .java and .class files in your "S:" drive. You can also get at these directories and files from anywhere on the Internet by using FileZilla (or other sftp client software) to connect to itsunix.albany.edu Use port 22 if you must specify a port.
DETAILS TO ADD THE BOOK CLASSES TO YOUR CLASSPATH IN DRJAVA

1. From DrJava’s Edit menu, select Preferences.

2. When the Preferences window opens, click the Extra Class Path: Add button (under the Resources tab).

3. In the list of files and directories shown in the new window should be your CSI201 directory. Click on that and then you will see the entry for the bookClasses directory. Click on that, and then on Select. Then click on OK at the bottom of the Preferences window. Make sure ONLY ONE extra Class Path remains.

4. Remember, you will have to do this only once on each computer you use.

DETAILS OF STEP 7

1. You’re done with DrJava for the lab. In the shell window, type the following command EXACTLY:

   /usr/local/depts/cs/geintro/verifyLab01

2. You must copy the last line of what that command printed on your Lab01 credit slip and give it to the TA for some of the credit for this lab. For maximum credit, it must report success. However, you may get as much help as much help from classmates as you need to make these happen on your lab computer.

3. When you are done in the lab, remember to log out with the command:

   logout

Unix commands introduced:

1. logging in, giving your NetId and typing your password with no echoing.
2. cd (change working directory) Note: Directory is the Unix word for "Folder"
3. pwd (print working directory)
4. mkdir (make a directory)
5. ls -l (list the contents of a directory with long details)
6. logout

DETAILS OF STEP 8 TRY THESE OUT TODAY, AFTER YOU LEAVE THE LAB

This is to make sure you know how to access files from your UAlbany S: drive so you can (1) transfer them (with ssh client software like FileZilla or sftp) to and from other computers, (2) use files from lab as starting points for project work and trying things outside of lab and (3) upload your work properly, as files with specific names, contents and structure, to Blackboard so this is not a problem when it comes time to submit project work that counts much more!

Instructions to access your S: drive in or out of lab: Navigate to https://wiki.albany.edu/display/public/askit/Personal+File+Storage+Space+(S-Drive)

Get some more practice with Turtle objects and their methods by trying more at home, your dorm or the UA Library. In the Library: Start DrJava under “DrJava”. You’ll have to download java-sources.zip from http://coweb.cc.gatech.edu/mediaComp-teach#Java, unzip and configure EXTRA CLASS PATH for bookClasses. Try some other shapes. How about a square centered at the current location? Hint: Use the penUp() and penDown() methods. Besides learning from the book what potential behavior Prof. Guzdial and Ericson programmed into their Turtles, go to our CSI201 Web site and click (near the bottom) to see the full documentation of the bookClasses.

Successful submission of your modified and perfected DrawWithMethodsApp.java and ArtisticTurtle.java file to the Lab01 assignment on Blackboard by Sept. 16 counts in your Lab grade. They MUST be uploaded as ATTACHMENTS (so we can retrieve them as files) NOT copy-pasted into Blackboard’s edit box!
Review Questions...may become iClicker questions in lectures the week after this Lab!

1. Match up each Unix shell command with its description:
   - `pwd` list working directory contents, one long line of details for each entry
   - `mkdir` make a new directory (fails if that directory already exists)
   - `cd` change the current working directory to the one you named after `cd` logout, so another person cannot delete your files.
   - `ls -l` print the current working directory
   - `logout` sign off so another person cannot delete your files.

2. Outside of lab, you will access your CSI201 files
   (A) Via email  (B) Can't.  (C) On the S: drive of Library computers (D) after you install and use FileZilla on your own computer (E) Using Blackboard, after giving your NetId, password and picking CSI201.  (F) Two answers: C and D  (G) Three answers: C, D and E

3. In this course, the only way or ways to submit programming work for grading and credit is or are:
   (A) Via email (B) Upload files to Blackboard as attachments (C) Copy and paste code into Blackboard (D) Filezilla (E) Make sure they are saved under your CSI201 directory so your TAs can access them.

4. How are you expected to keep the files you write in this course organized?
   (A) Keep all college work in one folder so all your college work is together in one place
   (B) Under CSI201 keep everything for CSI201 and NOTHING for any other purpose.  Also, each separate lab and project is in a separate sub-folder/directory under the folder/directory named CSI201.
   (C) Any way you like as long as it's all under CSI201

5. What does `System.out.println("Prof. S. Chaiken");` mean?
   (A) Print Prof. S. Chaiken right away.
   (B) Print Prof. S. Chaiken when and only when the program containing this statement is run.

6. What best describes programming?
   (A) Controlling what the computer does by typing commands with the keyboard rather than pointing and clicking.
   (B) Writing and saving instructions in the computer that might control what the computer does in the future.

7. Suppose you make a mistake in what you programmed.
   (A) When the program is run, the computer will do exactly what you programmed even if it is wrong.
   (B) The computer is smart enough to detect your mistakes and do the correct things anyway.

8. In this lab, what CALLS the method named `hook( ... )`?
   A) Code in the main method written in `DrawWithMethodsApp.java`
   B) Code in the `hook( ... )` method's defining body written in `ArtisticTurtle.java`

9. In this lab, what DEFINES the method named `hook( ... )`?
   A) Code in the main method written in `DrawWithMethodsApp.java`
   B) Code in the `hook( ... )` method's defining body written in `ArtisticTurtle.java`

10. When does the computer actually draw one hook?
    A) When he/she/it defines the hook method.
    B) When he/she/it CALLS the hook method.
public class DrawWithMethodsApp
{
    public static void main(String[] a)
    {
        System.out.println("Hello from main!");
        World wref = new World();
        ArtisticTurtle tref = new ArtisticTurtle(wref);
        //The 4 commands are supposed to draw a hook.
        tref.forward(50);
        tref.turn(90);
        tref.forward(35);
        tref.turn(-90);
        //Lab exercise:
        // (1) Remove or disable the above 4 non-comment commands.
        // (2) Edit those removed commands (by replacing tref by this)
        //     so they are programmed in the hook(param) method.
        //     This job was partially done for you and your job
        //     is to finish it.
        // The result should be that
        // INSTEAD of the hook drawing commands coming from main,
        // the 4 hook drawing commands are given THREE times
        // because you moved them into the hook(param) method
        // AND the code below calls hook(something) THREE times.
        tref.setPenColor(java.awt.Color.RED);
        tref.hook(3);
        tref.hook(1);
        tref.hook(2);
    }
}

-------------Lines below are the ArtisticTurtle.java file -------------
public class ArtisticTurtle extends Turtle
{
    public void hook(int numberParam)
    {
        System.out.println("The hook method has been called on ArtisticTurtle "+
                      this
                      +"with parameter value equal to "+
                      numberParam);
        this.forward(50);
        this.turn(90);
        //YOU have to finish the programming job by typing 2 more
        //java command lines below:
    }
    public ArtisticTurtle(World wrefParam)
    {
        super(wrefParam);
    }
    public static void main(String[] a)
    {
        System.out.println("DONT RUN ArtisticTurtle!!");
        System.out.println("Select DrawWithMethodsApp");
        System.out.println("and RUN that!");
    }
}