

# Eclipse & Silver Dollar Game

Wednesday, Sept 6, 2017

Lab 1  
CSC 062: Fall, 2017

---

## Eclipse

---

This laboratory is intended to get you started with the programs and systems in the lab, and to give you an opportunity to complete your first substantial program for the class. Be sure to read the description of the Silver Dollar Game on the next page before coming to class, and think about how you would solve it. The material on Setting up Eclipse is best read while sitting in lab.

Even though many of you have used **Eclipse** before, please follow the introductory steps below carefully. Correcting configuration errors is not impossible, but it is time-consuming and distracting. Before leaving the lab session, please practice turning in your code so that you will know how to do it with the regular assignment.

- 1. Setting things up:** You should all have your accounts and passwords. Log into one of the computers in the classroom. Among the icons in the dock at the bottom of the screen are four that will be important for the class.



Terminal



Eclipse



Safari

You will use **Terminal** and **Eclipse** today. **Safari** is a web browser (try it out!), through which you can read the course pages and documentation.

For now, open Safari and navigate to <http://www.cs.pomona.edu/classes/cs062/csc062.html> (if you are reading this on-line, you can just click on the url). Then click on the “+” to the left of where you typed the address to add it to your bookmarks. I suggest that you add it to the “Bookmarks bar” so that it will be easily accessible to you.

Open a terminal window and type the following commands. They will create a (private) folder for this class and a workspace within that folder.

```
cd ~/Documents
mkdir cs062
chmod 700 cs062
cd cs062
mkdir workspace
```

**Explanation:** The command `mkdir` makes a directory, and `cd` changes to that directory. The `chmod` command makes the directory private, so that only you can see its contents. A few other commands for your information: `ls` lists the contents of the current directory, `cd ..` changes the directory to one directory lower, `rm` removes a specified file and you can type `man <command>` for any command and it will give you more information (short for “manual”).

Note that much of this could have been accomplished using the mouse and interacting with menus. However, one goal for today is to get you used to using the terminal to accomplish tasks. All of the commands you are typing into the terminal window are UNIX commands, as the Macintosh operating system is a variant of UNIX (BSD UNIX, to be exact). These same commands will generally work with other flavors of UNIX, like LINUX. However, Windows computers use a different operating system, with slightly different commands to accomplish the same tasks.

Now start Eclipse. When it asks you for a workspace, select Browse, find the `/Documents/cs062/workspace` folder you just created, and click OK. With Eclipse still as the frontmost application, go to the menu at the top titled Eclipse and select Preferences. Then in the Preferences window, expand Java and then Build Path, and then select Classpath Variables. Click on New and enter the name BAILEY and the path

```
/common/cs/cs062/lib/bailey.jar
```

(or navigate to that file using your mouse, and select it.) These steps need only be done once.

2. **Adding libraries:** Eclipse works in units called “projects.” There will be one for each laboratory and each assignment in the class. To start a project, select File/New/Java Project. Give your project a name, like Lab1, but *do not click Finish yet!* Click Next, then Libraries, and then Add Variable. Add the variable BAILEY, and then click OK and Finish.

This step adds the bailey library (which we just setup) to our project so that our project knows that code is available. This allows us to use the `import` command in our code to bring outside resources (i.e. resources that are not included with java). For many of the labs, you will need to add the BAILEY variable to your project.

---

## Silver Dollar Game

---

1. Your program for lab today is to write a text-based version of the Silver Dollar Game. See Section 3.10 of *Java Structures* for a description of the game. Make sure you understand how the game is played before coding it up! Think about the methods described there and choose an appropriate data structure to model the game. This is a time for discussion in the lab; do not hesitate to ask questions or share your thoughts.

The behavior, in the console, will look like this:

```
_o____oo__ Next move? 6 4
_oo____o__ Next move? 2 2
Illegal move!
_oo____o__ Next move? 1 1
o_o____o__ Next move?
...
ooooo_____ You win!!
```

The pair of numbers after “Next move?” signifies the location of a coin and the number of squares that the coin is to move to the left. Remember that in languages like Java/C/C++ we start counting at 0 not 1, so the leftmost square is location 0, not 1.

For this first exercise, we have given you a start on the code for the class `TextCoinStrip`. It can be found at `/common/cs/cs062/labs/lab1/`. To get it in to your directory, copy the file `TextCoinStrip.java` into your `src` directory. To do this, type:

```
cp -r /common/cs/cs062/labs/lab01/* ~/Documents/cs062/workspace/Lab1
```

(where “Lab1” in the destination of the copy should be replaced by the name of your project for this lab). Then, go in to Eclipse and “ctrl+click” on the project and select “Refresh”. You should now see the starting code. (By the way, the `~` is an abbreviation for your home directory, and will be used a lot.)

Take some time to understand the code (I promise you’ll be much better off spending 5-10 minutes looking at the code before doing any coding yourself), then fill in the four missing methods.

- `toString` which creates the string representation of the strip,

- `isLegalMove` which determines if a move is legal,
- `makeMove` which makes a (legal) move, and
- `gameIsOver` which determines if the game is completed.

---

## Submitting Your Work

---

The next part will have to be repeated every time you want to turn in a program.

- From within Eclipse, select “Export” from the “File” menu.
- Click on the triangle next to “General” in the dialog box. Select “File system” and click next.
- Make sure all the files in the dialog on the right hand side are checked, and then click the “Browse” button next to the “To directory:” entry
- Select “Desktop” from the pulldown menu and click on “Open”.
- Click “Finish”. This should create a new folder on the desktop. The folder name should be the same as your Eclipse project name. Inside the folder, there should be a *src* folder (with your .java files) and a *bin* folder (with your .class files).
- Quit Eclipse.
- Rename this folder `LabNN_LastNameFirstName` or `AssignmentNN_LastNameFirstName` according to whether it is a lab or an assignment. NN is a two-digit integer indicating which lab (or assignment) this is, and you should use your actual first and last names.
- Fill out the `labNN.json` or `asnNN.json` file that came with the assignment and copy it into the exported directory at the top level. You should do the following:
  - (a) Put your Pomona CS username (the one you use to log on to the lab computers), enclosed in double quotes, inside the “collaborators” list (in the square brackets; you can replace the “yourusername” that’s there). If you worked with a partner, replace the “yourpartnersusername” with their name, otherwise delete that entry.
  - (b) If you attempted extra credit for the assignment, change the “false” to a “true” on the “ec” line.
  - (c) If you have any comments to the graders, add them in the “notes” field (put them between the double quotes).
- Submit the folder you just created using the submit script.

The script is located at `/common/cs/cs062/bin/submit` on both the lab computers and remotely-accessible machines (e.g., `little.cs.pomona.edu`). You need to give the submit script the class and the name of the assignment you are submitting (each lab and assignment should tell you what to submit as) along with the directory you want to submit. *The directory you are submitting must exist on the Pomona CS servers—either on a lab computer or copied to a remotely-accessible computer.*

2. The command will look something like this (assuming you followed the directions above; type this into the Terminal app):

```
/common/cs/cs062/bin/submit cs062 lab01 ~/Desktop/Lab01_PapoutsakiAlexandra
```

3. **If the submit script does not print out “SUCCESS” then it did not work!** Read the error message that it printed and try to correct it. For example, you may be missing a required file (all files are specified relative to the top-level directory that you just created on your desktop).
4. Submitting correctly is your responsibility, and if you forget to submit or submit unsuccessfully without following up, everyone will be unhappy.

## 5. More Unix

You will find it very helpful to become more expert in UNIX commands. To find out more, go to <http://www.dci.pomona.edu/index.php?f=Docs&id=unixhelp>

**6. Extension:** This is a suggestion for those who want to go beyond today's exercise; it is not a course requirement. Let the computer assume the role of one player and experiment with strategies for the Silver Dollar Game. You may conduct your experiments in either the text-based or the graphical world. Here are some simple strategies:

- Move the leftmost possible coin as far as possible.
- Move the rightmost possible coin one square.
- Find a coin capable of moving the farthest and move it the full distance.
- A "gap" is a sequence of unoccupied squares between two coins. If possible, make a move to increase the number of gaps.