Programming Assignment 1

Due: 11:59pm, Tuesday, October 6
Worth 6% of final grade

Overview
The goals of this assignment are to:
1. Get you started using the Unix environment
2. Focus on programming style and semantics
3. Debug programs
4. Understand APIs
5. Learn how to turn in your programs

Reading
Review the course syllabus, esp. details on slip days.

Setup
Look up your CSE 8B account. You’ll use this account throughout the quarter to submit your assignments.

Open a new Linux terminal window. In your home directory, create/make a new directory called HW1 and change directory into that directory (in Unix, the tilde ~ expands to your home directory):

```
$ mkdir ~/HW1
$ cd ~/HW1
```

Copy the contents of /home/linux/ieng6/cs8b/public/HW1/* to your HW1 directory via the command:

```
$ cp /home/linux/ieng6/cs8b/public/HW1/* .
```

This command copies all contents of /home/linux/ieng6/cs8b/public/HW1 to the specified directory. Note the use of the star wildcard, *, which can represent zero characters, all single characters, or any string, and the single dot means your current working directory. You should now have 3 files in your HW1 directory:

```
$ pwd
/home/linux/ieng6/cs8b/<username>/HW1
$ ls
   Debug.java  Animate.java  objectdraw.jar
```
Part I: Debugging (10 pts)
If you attempt to compile the program Debug.java via:

```
$ javac Debug.java
```

You’ll get a few programming errors. Open the file Debug.java with emacs or vim and find and fix each of the errors. For each error, comment out the original line and replace with the corrected line. Make sure your program now compiles and runs via:

```
$ javac Debug.java
$ java Debug
Enter your name: Adam Jundt
Your name is: Adam Jundt
```

Also, for each .java file be sure to include your name, student ID, and username.

Part II: Reading an API (30 pts)
In lecture 2 we briefly discussed the objectdraw library. You can find the API (Application Program Interface) online at http://eventfuljava.cs.williams.edu/library/objectdrawJavadocV1.1.1/index.html. Edit the program Animate.java to draw an image that has some kind of animation, e.g. sunrise/sunset, bouncing balls, a movie scene. At the minimum, there must be some kind of object/shape that moves around the screen. You will be graded on a range of creativity, animation, and complexity.

To do this, you’ll need to look through the API and find which available classes and methods will help you. The starter file, Animate.java, has comments that will show you where to get started. You can compile and run your code via:

```
$ javac -cp objectdraw.jar:. Animate.java
$ java -cp objectdraw.jar:. Animate
```

**Note:** to exit the program you’ll need to return focus to the terminal and hold CTRL-C. If you’re working remotely and trying to visualize your animation, you’ll find that it’s very choppy. This is due to network lag. You’ll want to either develop on your laptop with the JDK installed or on the lab machines.

**Grading:** You’ll be graded based on the following rubric:

<table>
<thead>
<tr>
<th>Category</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creativity. Is it interesting to watch</td>
<td>1-10</td>
</tr>
<tr>
<td>Animation. The way objects move. Do they respect borders.</td>
<td>1-10</td>
</tr>
<tr>
<td>Complexity. How much time was invested in the program.</td>
<td>1-10</td>
</tr>
</tbody>
</table>

You’ll also need to step through your animation with your tutor and point out sections of your code. You’ll need to be able to do this using either emacs for vim for credit.
**Best Animation**

Tutors have the option of selecting 1 animation to represent their group. We will present the top 5 animations to the class for voting (via clickers) and the winner will get an automatic 100% on the final. The winning team’s tutor will receive a gift card. The remaining 4 animators will receive 1% extra credit added to their final grade.

**Style Requirements (10 pts)**

You will be graded for the style of programming on this assignment. A few requirements for style are given below and at [https://google.github.io/styleguide/javaguide.html](https://google.github.io/styleguide/javaguide.html). These guidelines for style will have to be followed for all the remaining assignments. Read them carefully. In the template code provided below for this assignment, all of these requirements are met (replace comments appropriately).

- Use reasonable comments to make your code clear and readable.
- Use Javadoc style comments for all classes and methods.
  - The comments should describe the purpose of your program and methods.
- Use reasonable variable names that are meaningful.
- Use static final constants to make your code as general as possible. No hardcoding constant values inline.
- Judicious use of blank spaces around logical chunks of code makes your code much easier to read and debug.
- Keep all lines less than 80 characters. Make sure each level of indentation lines up evenly.
- Every time you open a new block of code (use a '{'), indent farther. Go back to the previous level of indenting when you close the block (use a '}').
- Always recompile and run your program right before turning it in, just in case you commented out some code by mistake.

**Turnin Instructions**

Remember the deadline to turn in your assignment is Tuesday, October 6, by 11:59pm. Make sure the program works correctly in your cs8b login on the workstations in CSE B260. When you are ready to turn in your program in, type in the following command and answer the prompted questions:

```bash
$ cd ~/HW1
$ bundleP1
Good; all required files are present:
    Debug.java
    Animate.java

Do you want to go ahead and turnin these files? [y/n]y
OK. Proceeding.

Performing turnin of approx. 6416 bytes (+/- 10%)```
Copying to /home/linux/ieng6/cs8b/turnin.dest/cs8bezz.P1
...
Done.
Total bytes written: 6656
Please check to be sure that's reasonable.
Turnin successful.

bundleP1 will look for the 2 files: Animate.java and Debug.java and will zip the files into a single turnin submission. You can turn in your program multiple times. The turnin program will ask you if you want to overwrite a previously-turned in homework. ONLY THE LAST TURNIN IS USED!