

Lab 06: The Cow Says...

Overview

This lab is designed to introduce students to the Bash Command Line Interface (CLI) and the concept of CLI arguments and give them practice writing classes. For (only) this lab, you are not allowed to use any windowed editor (such as IntelliJ, Eclipse, or Notepad++). Instead, you will use a Unix-based command line editor.

The **cowsay** utility is a popular Unix program from the 20th century (see <https://en.wikipedia.org/wiki/Cowsay>). You will write a slightly simplified **cowsay** program that takes in several arguments and prints out different text depending on the arguments. You must log in to the school computers using your CISE account to do this lab.

Tools

Please note that **you must use a text editor and the terminal to edit and run your program and its directories**. It is advised students learn/review basic Unix shell commands before beginning; a good run-through can be found here: <https://linuxjourney.com/lesson/the-shell>.

Follow these steps to get started on the lab:

- 1) Open a terminal and enter the **pwd** command to identify the path to the working (current) directory (folder)
- 2) Enter **ls** to list the contents of the current directory
- 3) Use the **mkdir** command to make a new directory called *CowLab*.
- 4) Use **ls** to see the change, then **cd** to change to the directory *CowLab*.
- 5) Do your lab work in that folder. Use your googlegfu skills to find more commands.

Recommended text editors include **nano** and **joe**. (If you hate yourself, you can use **vi** or **emacs** too.) You'll also need to use the **javac** program for Java compilation and the **java** command to run Java programs.

You can read more information about some of these commands here:

<https://www.howtogeek.com/howto/42980/the-beginners-guide-to-nano-the-linux-command-line-text-editor/>
<https://introcs.cs.princeton.edu/java/15inout/linux-cmd.html>

History of Command Line Names (Just for Fun)

In case you are wondering: **nano** is a free-software version of a program called **pico**, which is the stand-alone editor for Pine at the command line. The **p**ine email program was an alternative version of Elm – you see, Pine Is Not Elm. The elm program was an earlier email manager. (If you think my jokes are bad, you cannot fathom the depths of terrible humor present in computer science and engineering! This is but the beginning of a terrible humor journey.)

Specification

Students will construct two classes: a driver class with a **main()** entry point (**Cowsay**) and a data class (**Cow**). Note that **HeiferGenerator.java** is provided for you; your code must use this class to create the cow objects. All attributes / methods must be private unless noted in the specification!

Provided For Students - HeiferGenerator

```
public static Cow[] getCows()
```

Returns an array of cow objects from the built-in data set. This will call the `Cow` constructor and `setImage()` methods of the cow class to properly initialize new cow objects uniquely for each data set.

Cowsay Class (Program Driver)

Your program must accept command line arguments. Command line arguments are captured as part of the `args` variable passed into the `main()` method of the driver class. (Review lecture slides for examples!)

The command line arguments that must be supported are as follows:

java Cowsay -l	Lists the available cows
java Cowsay MESSAGE	Prints out the MESSAGE using the default COW
java Cowsay -n COW MESSAGE	Prints out the MESSAGE using the specified COW

If a user calls for a cow that does not exist, the program should print out “Could not find *[COWNAME]* cow!”

Output Samples



Suggested Methods

The following methods are suggested to make development easier, but are not required:

```
private static void listCows(Cow[] cows)
```

Displays the list of available cows from an array of **Cow** objects.

```
private static Cow findCow(String name, Cow[] cows)
```

Given a name and an array of **Cow** objects, return the **Cow** object with the specified **name**.

Cow Class

The **Cow** class facilitates the creation and use of cow objects by providing the following methods (which students must implement):

```
public Cow(String name)
```

This method should be the **only** constructor for this class. There should not be a default constructor!

```
public String getName()
```

Returns the name of the cow.

```
public String getImage()
```

Returns the image used to display the cow (after the message).

```
public void setImage(String image)
```

Sets the image used to display the cow (after the message).

Testing

Run the **cow_test.sh** script provided. To run it, place it in the same folder as the rest of your files and make it executable via **chmod +x cow_test.sh**. Then run it: **./cow_test.sh**. This script contains a series of commands to run **Cowsay** in various ways, streaming the results into the text file **output.txt**. It then uses **less** command to view the output. Your output file should match the sample output in this document.

Submissions

NOTE: Your output must match the example output **exactly**. If it does not, *you will not receive full credit for your submission!*

Files: Cowsay.java, Cow.java, HeiferGenerator.java
Method: Submit on Canvas

Sample Output

```
>javac Cowsay.java
>java Cowsay Hello World!
```

```
Hello World!
      \
       ^__^
      (oo)\_______
      (_____)\/    )\/\
          ||----w |
          ||     ||
```

```
>java Cowsay -n kitteh Hello World!
```

```
Hello World!
      \
       ("`-.' ' -/'") ._____.--.' ' " -.'
          \ * * ) .--. ( ) .--. `
          (Y_.) ' . ) `.-; .--.'
          _..--'--'--' /--'-'',4
      ( i l ),-' ' ( l i),' ( ( ! .-'
```

```
>java Cowsay -l
Cows available: heifer kitteh
```

```
>java Cowsay -n ninja Hello world!
Could not find ninja cow!
```

```
>java Cowsay Hello -n kitteh
```

```
Hello -n kitteh
      \
       ^__^
      (oo)\_______
      (_____)\/    )\/\
          ||----w |
          ||     ||
```