

---

**DUE DATES:**

Monday recitations: 9:00 PM on 10/2  
Wednesday recitations: 9:00 PM on 10/4  
Thursday recitations: 9:00 PM on 10/5  
Friday recitations: 9:00 PM on 10/6  
Saturday recitations: 9:00 PM on 10/7

---

**Ready!**

In this lab you will use the Java you have learned in lecture to write a small program. In particular you will:

- define classes
- declare local variables
- declare instance variables (**NEW!**)
- create objects
- assign values to local variables
- initialize instance variables in a constructor (**NEW!**)
- call methods
- define methods (**NEW!**)

If you are unsure how to do these things you should review your lecture notes, slides posted on the course website, and readings from the textbook.

---

**Set!**

1. Log in
2. Start Eclipse
3. Switch to the CVS Repository Exploring perspective
4. Check out the CSE115-Lab3 project from the Labs repository
5. Switch to the Java perspective

---

**Go!**

In this lab you will write several small programs which, when run, create one or more BarnYards and add critters to them. Do all three parts. Part 1 begins on the next page.

---

### Part 1

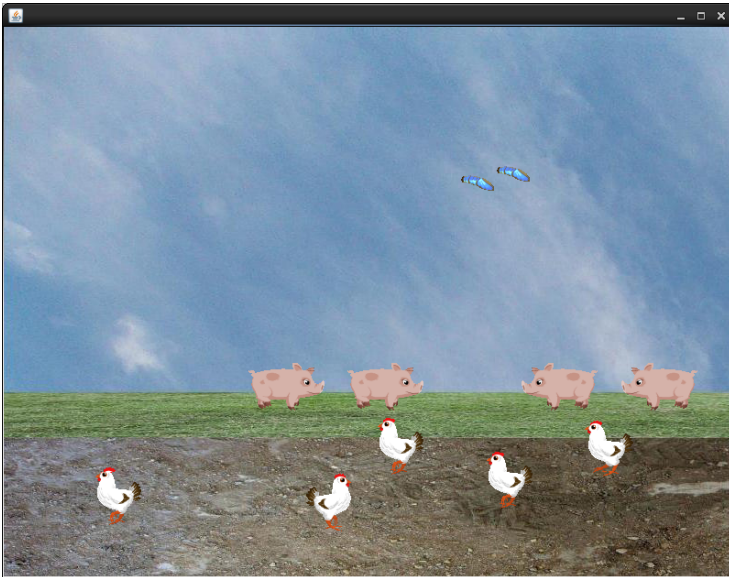
Edit the definition of the Farm class in the part1 package to do the following:

- Declare an instance variable of type `example1.BarnYard`.
- Initialize the instance variable to a new `example1.BarnYard` instance.
- Define a void and parameterless method named `addOneMovingChicken` which creates an `example1.Chicken` object, adds it to the `BarnYard`, and starts it moving.
- Define a void and parameterless method named `addOneMovingPig` which creates an `example1.Pig` object, adds it to the `BarnYard`, and starts it moving.
- Define a void and parameterless method named `addOneMovingButterfly` which creates an `example1.Butterfly` object, adds it to the `BarnYard`, and starts it moving.

Edit the definition of the Tester class in the part1 package to do the following:

Create, in the constructor, a new `part1.Farm` object and call the `addOneMovingChicken` method five times (thereby adding five moving Chickens to the Farm's `BarnYard` object), call the `addOneMovingPig` method four times (thereby adding four moving Pigs to the Farm's `BarnYard` object), and finally call the `addOneMovingButterfly` method twice (thereby adding two moving Butterflies to the Farm's `BarnYard` object),

You are expecting that a window will open on the screen, displaying an image inside it, like this:



You might recognize this picture - it is from part 5 of lab 2. You've now accomplished the same thing you did at the end of lab 2, but with a bit more code. With the Farm class defined in this way we can go on (in parts 2 and 3) to do some more interesting things!

---

### Part 2

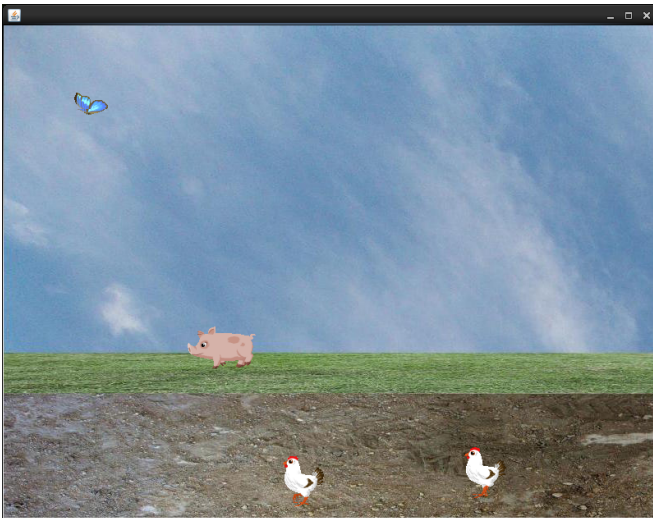
In this part you will use two Farm objects to create a `MidSizedFarm` object. The special property of the `MidSizedFarm` is that when animals are added to it they alternate which of the two Farms they go into.

Edit the definition of the `MidSizedFarm` class in the `part2` package to do the following:

- f) Declare two instance variables of type `part1.Farm`. Let's call them `_farm0` and `_farm1`.
- g) Initialize each of the instance variables to a new `part1.Farm` instance.
- h) Define a void and parameterless method named `addChickenAlternating` which adds a moving chicken to the Farm referred to by `_farm0` (by calling the `addOneMovingChicken` method), and then interchanges the values of `_farm0` and `_farm1`. Your TA will explain how to do this. [For those with some prior programming experience: you may NOT use any kind of conditional.]
- i) Define a void and parameterless method named `addPigAlternating` analogously.
- j) Define a void and parameterless method named `addButterflyAlternating` analogously.

Edit the definition of the `Tester` class in the `part2` package to do the following:

Create, in the constructor, a new `part2.MidSizedFarm` object and call the `addChickenAlternating` method three times, then the `addPigAlternating` method two times, and finally call the `addButterflyAlternating` method three times.



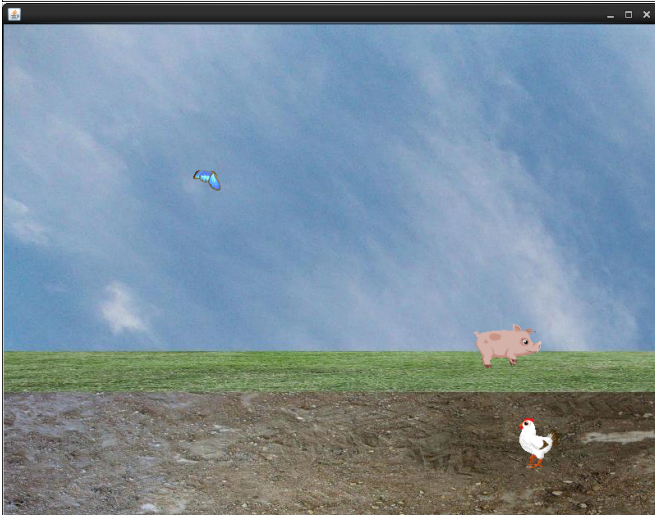
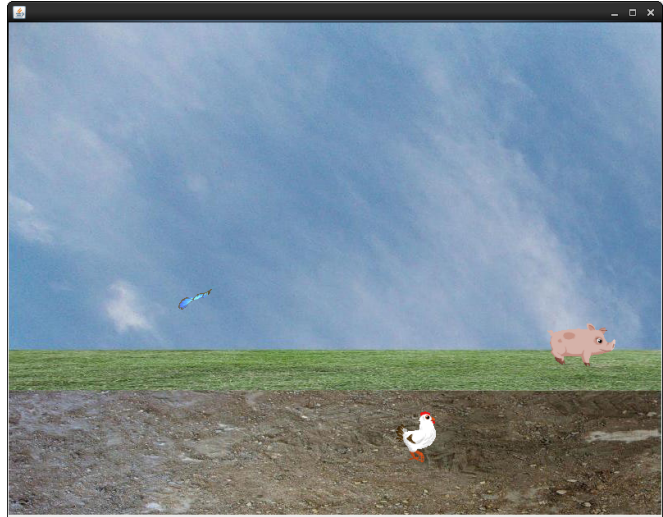
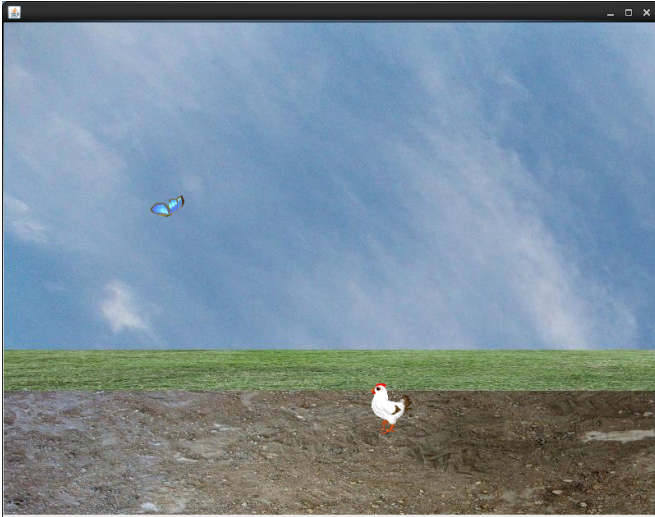
### Part 3

In this part you will use three `Farm` objects to create a `MegaFarm` object. The special property of the `MegaFarm` is that when animals are added to it they rotate through which of the three Farms they go into. (You can think of the `MidSizedFarm` as rotating through the two Farms.)

Edit the definition of the `MegaFarm` class in the `part3` package to do the following:

- k) Declare three instance variables of type `part1.Farm`. Let's call them `_farm0`, `_farm1`, and `_farm2`.
- l) Initialize each of the instance variables to a new `part1.Farm` instance.
- m) Define a void and parameterless method named `addChickenRotating` which adds a moving chicken to the Farm referred to by `_farm0` (by calling the `addOneMovingChicken` method), and then rotates the values of `_farm0`, `_farm1` and `_farm2`. Your TA will explain how to do this. [For those with some prior programming experience: you may NOT use any kind of conditional.]
- n) Define a void and parameterless method named `addPigRotating` analogously.
- o) Define a void and parameterless method named `addButterflyRotating` analogously.

Edit the definition of the `Tester` class in the `part3` package to do the following:  
Create, in the constructor, a new `part3.MegaFarm` object and call the `addChickenRotating` method three times, then the `addPigRotating` method two times, and finally call the `addButterflyRotating` method three times.



---

### Submitting your project to Web-CAT

Make sure you submit your work on time; due dates are listed at the beginning of this lab description. This lab will be automatically graded by Web-CAT, using not only the code you wrote but also some additional tests that your instructor wrote. You may submit as many times as you wish. Your last submission is the one that counts (so consider carefully whether you want to make any late submissions, as the late penalty is 20 points per day or portion thereof late).

---