Object, Classes, Methods

- Classes define
  - the state (data), usually private
  - behavior (methods) for an object, usually public
- There can be many objects created based on a class.
- Method – sequence of instructions that access the data of an object
  - Accessor – access, don’t change data
  - Mutator – changes the data

Example - class Chicken

- State
  - weight, height, name
- Behavior (methods)
  - Accessor methods
    - getWeight, getHeight, getName
  - Mutator methods
    - eat – adds weight, adds some height if under 12.0
    - sick – lose weight
    - changeName
Constructing Objects - new

- Create three chickens
  - “Fred”, weight 2.0, height 3.8
  - “Sallie Mae”, weight 3.0, height 4.5
  - “Momma”, weight 6.0, height 8.3
- Use Chicken constructor
  Chicken one = new Chicken(2.0, 3.8, "Fred");
  Chicken two = new Chicken(3.0, 4.5, "Sallie Mae");
  Chicken three = new Chicken(6.0, 8.3, "Momma");

Object References
- Variable of type object – value is memory location

Parts of a Class
- State
  - Data
- Constructors
  - Initialize state when object is created
- Accessor methods
  - Accessing data
- Mutator methods
  - Modify data – change the state

one = two;
- Now they reference the same object

```java
System.out.println(one.getName() + " has "+
  one.getName().length()+ " letters.");
System.out.println(two.getName() + " has "+
  two.getName().length()+ " letters.");
```
Class Example

- **Chicken class** – `Chicken.java`
  - Defines state and behavior of Chicken
- **Farm class** – `Farm.java`
  - Creates Chickens with “new”
  - Invokes the Chicken constructor
  - Calls chicken methods on Chickens to access or change state

What happens here? How many Chickens are constructed?

```java
Chicken x, y;
Chicken z = new Chicken(1.0, 2.1, "baby");
x = new Chicken(10.3, 8.1, "ed");
y = new Chicken(6.2, 6.3, "mo");
Chicken temp = x;
x = y;
y = temp;
z = x;
```

Generating Random Numbers

- `java.util.Random` class in Java library
  Random generator = new Random();
- Methods:
  - `nextInt(int n)` - returns integer from 0 up to but not including n
  - `nextDouble()` – returns double between 0.0 and 1.0
    ```java
    int num = generator.nextInt(10);
    ```

Examples

- Go over `Chicken.java`, `Farm.java`
- Go over `Skier.java`, `SkiRace.java`, `RunSkiRace.java`