Objects, Classes, Methods

- Classes define
  - the state (data), usually private
  - behavior (methods) for an object, usually public
- Many objects can be created based on one class.
- Method – sequence of instructions that access and/or manipulates the data of an object
  - *Accessor* method – access, don’t change data
  - *Mutator* method – 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()`

- **Syntax Note:** in code method name always followed by parentheses
Method Features

- **Return values**
  - Methods can return information
    - Accessor methods require that
    - Have return type in header specifying type of info
    - Use: `w = chick.getWeight();`

- **Parameters**
  - Methods may receive information thru parameters
    - Mutator methods usually require that
    - Method header includes parameter definition in parentheses
    - Use: `chick.newName("Elsa");`

- **May have both parameters and return values**
Constructing Objects - new

- **The new operator**
  - Used to create objects

- **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**
  
  ```java
  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");
  ```
Primitive Types

- Java builds in *primitive types* for dealing with numbers
  - Eight primitive types; only use few at first
    - **int** - holds whole numbers only
    - **double** – can deal with fractions
      - think scientific notation
  - These are *not* objects
    - They have *no methods*
  - Constants:
    - **int** – no decimal point: 256
    - **double** – contains decimal point: 3.14159265
Another Class - String

- Use the API methods for String class
  - To get to API from CompSci 6 web page
    - Click on “resources”
- Print length of Chicken one’s name.
- length() is a method in String class.

```java
System.out.println(one.getName() + " has 
                   + one.getName().length() + " letters.");
```
Object References

- Variable of type object – value is memory location

```
one = 

two = 
```

```
Chicken

weight = 2.0
height = 3.8
name = "Fred"

Chicken

weight = 3.0
height = 4.5
name = "Sallie Mae"
```
Assign: `one = two;`

- Now they reference the *same* object