CS115 Week 4
Methods/Variables, Constructors, Strings, Loops, Object Interaction

BRING A FLASH DRIVE WITH 2 FILES:
www.cs.iit.edu/~cs115mb/Counter.java
www.cs.iit.edu/~cs115mb/counter.png

New Scenario
A random number of balls of random colors moving at random speeds and starting in random directions. Every bounce off a wall changes the color and counts on a counter.

Constructors
Default Constructor – no arguments, sets any object attribute to some default value. If you do not write any constructor, you will get a default constructor automatically.

Non Default Constructor – arguments, call “set” methods (called mutators) to verify each argument and set object attribute.

Inherited classes first run the superclass default constructor (or code a call to the superclass non-default constructor).

Instance Variables (object attributes)
How do you know what attributes an object should have?
Whatever uniquely defines an object. Only attributes that “act()” method may need to use. Not attributes that are derived from other attributes. Not attributes that are common to all objects of the Class (use Class variable instead). Think about what attributes are inherited.

Class Variables
- Shared by all instances (objects) of the class
- Use the “static” keyword
- Constants (instance or class) – use the “final” keyword, usually all capitals for identifier name
String

- Strings are delimited by "hello world"
- Strings can be concatenated (put together) by using the + operator
- String is a Class in the java.lang package with many String manipulation methods. No special import is needed (automatically imported).
- You do not have to call a constructor to create a String object, you can just assign. String name = "Matthew Bauer";

Loops

- A loop is a repetition control structure
- It causes a single statement or block to be executed repeatedly while an expression is true
  "while" loops (now)
  "for" loops (later)

While Statement

```java
while (Expression)
{
    // loop body
}
```

NOTE: Loop body can be a single statement, a null statement, or a block

Object References (to support object interaction)

- identifier of the object, sometimes called the pointer or handle
- Allows multiple objects to call methods on the same object instance
- If all object reference are deleted (or set to null) the object is no longer available and thrown away.