CSE 113 A

September 21 – 25, 2009

Announcements

- No classes held on Monday, 9/28 until 6:00pm – university holiday.
- Lab 1 due 10/2
- Exam 1 10/7

Documentation

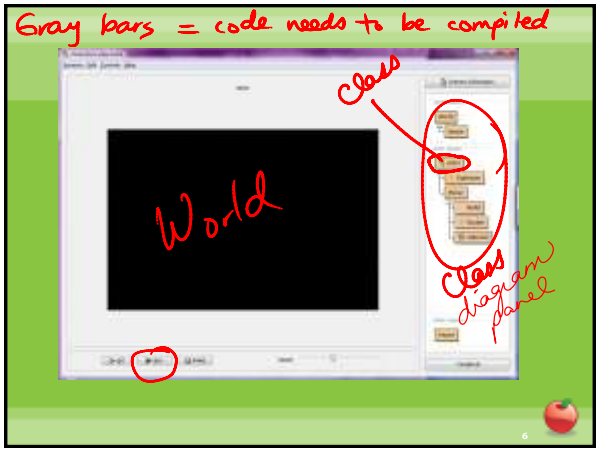
- Inside of Greenfoot, you can view the documentation about the built-in Greenfoot classes. Find this option under the Help menu.
- The documentation can help you better understand how to use certain methods from the built-in classes.

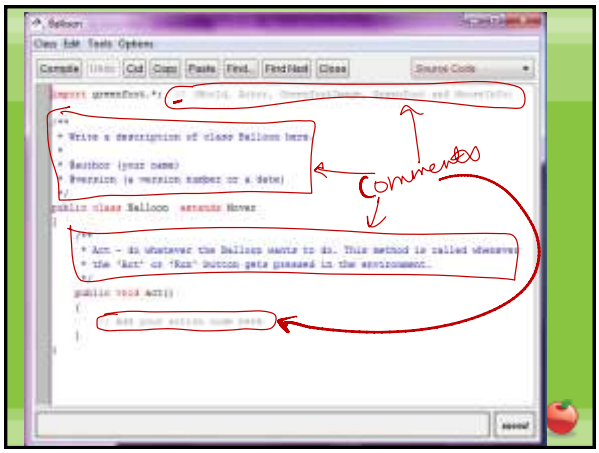
Strings

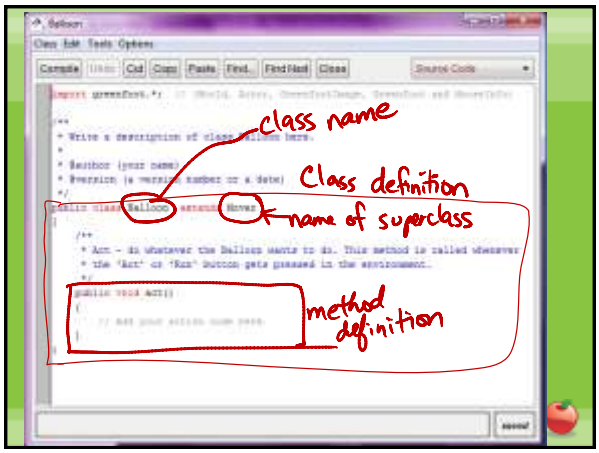
- Strings are a built-in type (object) inside of Java.
- Strings are a sequence of letters, digits, or other characters.
- If you want to specify a String literal, you need to surround it in quotes.
  - "this"
  - "a"
  - "left"

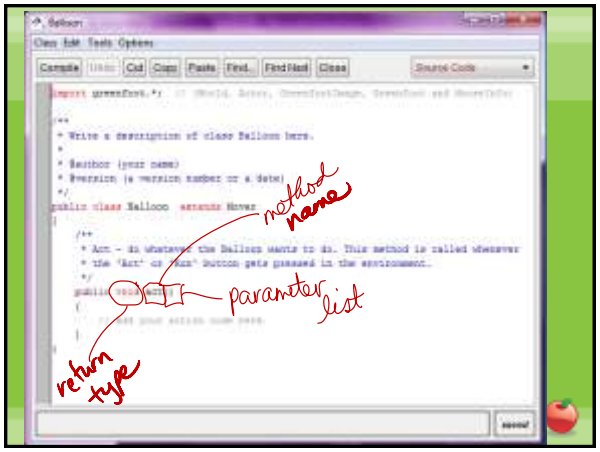
Review

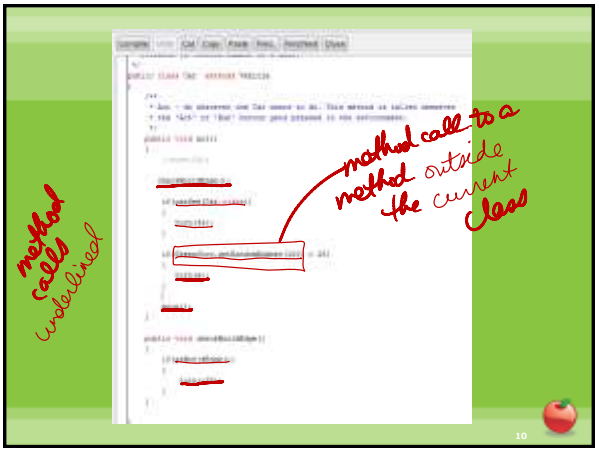
- The next several slides indicate review materials that were covered in class on Monday 9/21 and Wednesday 9/23. They incorporate the main ideas from Chapter 1 – 3 of the text.











Write the code for an act method that does the following:

- if hit edge of world, turn between -30 and 30 degrees
- if hit Car, play sound "crash.wav" and stop scenario
- 25% of time - move
- 50% of time - turn 5°

```
if(atWorldEdge())
{
  turn(Greenfoot.getRandomNumber(60)-30);
}
if(canSee(Car.class))
{
  Greenfoot.playSound("crash.wav");
  Greenfoot.stop();
}
```

```
if(Greenfoot.getRandomNumber(100)<25)
{
  move();
}
if(Greenfoot.getRandomNumber(100)<50)
{
  turn(5);
}
```

13

Questions

- Use the previous slides as a study guide. The answer for the last question posed on the slides will be available the week of September 28th.

14

Constructors

- Constructors are special methods that are called each time an instance of a class is created.
- Constructors inside source code:

```
public SameNameAsClass()
{
}
```

- Note that there is no return type and the constructor will always have the same name as the name of the class.

15

Constructors

- Constructors are special methods that are called each time an instance of a class is created.
- Constructors inside source code:

```
public SameNameAsClass()
{
}
```

- Note that there is no return type and the constructor will always have the same name as the name of the class.

16

Constructors

- Inside the body of the constructor (inside the {}), you can do any of the same things you can do inside of other methods.
- Therefore, we can call methods from within a constructor.
- In our example, we call

```
super(560,560,1);
```

- This is a call to a method named **super**. **super** is a keyword that actually indicates a call to the superclass' constructor.

17

Adding Objects to the World

- Note that the **addObject** method of the world takes as its first parameter an Actor to be added.
- We need to create an actual instance to pass into this method.
- To create an object inside Java source code:

```
new ConstructorName();
```

- **new** is a keyword indicating that we are creating a new instance.
- **new** is followed by a call to the class' constructor. Values are inserted in the () if needed.

18

Adding Objects to the World

- **addObject** also takes an x and y coordinate as parameters.
- We need to remember that in the coordinate system for graphics on computers, origin (0,0) is the upper left hand corner.
- The values of x increase as we move right on the screen and the values of y increase as we move down on the screen.

19