Announcements

• Assignment 3 storyboard due today!
• Assignment 3 worlds due Thursday
  – Turn worlds in on Blackboard
  – Name textfile “README” or README.txt
  – Put all three files in a folder named assign3LOGINID
    where LOGINID is your Duke login
What we will do today

• Chapter 4 Sec 2 – Parameters
• Classwork
Overview

• The need for more flexible methods
• Creating methods with parameters
• Passing arguments to methods
• Demo
  – Using Alice – methods with parameters
A Beetle Band

- Create an animation for a bug band as an ad for their next concert
- Each band member will perform a short solo
Storyboards

• Each bug member will perform a solo

Do together
  Do in order
  georgeBeetle move up
  georgeBeetle move down
  play sound

Do together
  Do in order
  ringoBeetle move up
  ringoBeetle move down
  play sound

Do together
  Do in order
  paulBeetle move up
  paulBeetle move down
  play sound

Do together
  Do in order
  lennonBeetle move up
  lennonBeetle move down
  play sound

• Note: instruments are already vehicled to the band member playing them.
Code for georgeBeetle

- We will need four versions of the code, one for each band member
- This code will only work for georgeBeetle
A Better Solution

• Four versions of similar code is tedious.
• The things that change are
  – The beetle
  – The music the beetle plays

• Better Solution: write a more flexible method
Parameters

- Built-in methods provide **flexibility** by providing parameters such as distance and duration
- Parameters allow you to pass in values
  - The values are **arguments**
- **Example:**

  ![Parameter Example](image)

  What are the parameters?
  What are the arguments?
  How many of each?
Types of Parameters

• Alice provides several types of parameters that can be used in your methods.
The Storyboard

- Write one method and use parameters for
  - Which bandmember to perform
  - Which music to play

Solo:
Parameters: `bandMember, music`
Do together
  Do in order
    `bandMember` move up
    `bandMember` move down
  play `music`
Creating a Method Named Solo

• Can be used for any band member
• Will need
  – An Object parameter – which band member is to play a solo
  – A Sound parameter – which music should be played
World.solo with parameters
Calling the solo method

- Note that in each call, arguments must be given for both parameters.
A Number Parameter

- Add a number parameter for **height** the **bandMember** jumps up and down

Must add this argument to each call.

**Demo**

<table>
<thead>
<tr>
<th>World.solo</th>
<th><code>bandMember = georgeBeetle</code></th>
<th><code>music = World.bassGuitar</code></th>
<th><code>height = 2</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>World.solo</td>
<td><code>bandMember = lennonBeetle</code></td>
<td><code>music = World.guitarSolo</code></td>
<td><code>height = 1</code></td>
</tr>
<tr>
<td>World.solo</td>
<td><code>bandMember = ringoBeetle</code></td>
<td><code>music = World.DRUM</code></td>
<td><code>height = 0.5</code></td>
</tr>
<tr>
<td>World.solo</td>
<td><code>bandMember = paulBeetle</code></td>
<td><code>music = World.saxophone</code></td>
<td><code>height = 4</code></td>
</tr>
</tbody>
</table>
Classwork Today

• Create worlds
  – Beetle band duet
  – Frog Escape