Where are we?
atories

Our language is a “toy” language.

- **Fundamentals** are exactly like “real” languages.
- **Concepts of parameters, reuse, design** are classic.
- First half of the semester, we talked about problems scientists and particularly computer scientists solve.
- Second half, we look at the fundamental tools used in solving those problems.

Computer Science – is it for you?

- If you have a real talent for this.
- Computer Science degrees earn $51K starting and over $100K after a few years.
- There are jobs now and only getting more!
- Projected to increase by 38 percent over the next ten years, which is much faster than the average for all occupations. This occupation will generate about 324,000 new jobs, over the next decade, one of the largest employment increases of any occupation.
- Try CS 1400 next semester.

Chapter 13:
Methods, Functions, and More about Variables

Starting Out with Alice: A Visual Introduction to Programming
First Edition
by Tony Gaddis

Writing Custom Class-Level Methods

- **Class-Level Methods**
  - Methods that are part of a class
- **Create an instance** of the class
  - Select the instance
  - Create new method
  - Enter a name for the new method (remember to use camelCase)

Try one

- **Takes any object and makes it jump**
- **What is input?**
Who “owns” the method?

- You can also have ‘world’ methods – that belong to the world.
- The method should be part of the most ‘logical’ group.
- If you make ‘disappear’ part of the world, when the magician is moved to another world, he/she won’t know how to make objects disappear.

Reuse – means creating to be used again!

Saving an Object to a New Class

- Custom methods are for only a single instance of the class. If you “copy” the instance, the new instance will also have the methods. If you create a new instance or add to one of two instances, they don’t share methods.
- If methods will be needed in other worlds, need to create a new class.
- Saving as a new class, adds the new method to that object permanently.
- For example, if you get a chicken to walk, you might want to allow the class to be used by others (and yourself)
- Programming becomes very creative.

The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination. Few media of creation are so flexible, so easy to polish and rework, so readily capable of realizing grand conceptual structures...

Frederic Brooks, Jr

Inheritance

- New methods were added and a new class was created.
- The new class still contains the methods that had previously been associated with the original object and class (such as Move, Turn, Say)
- The new class also contains the properties that had previously been associated with the original object and class (such as Color, Opacity, Vehicle)
- The new class inherited all the original methods and properties from the original class

Stepwise Refinement

Better to start out with general idea and then develop the detail

- Refined to have more detail added to it
- Algorithm becomes several methods

Stepwise Refine Example

Note the reuse

Original Algorithm:
Raise the right leg
Loop 10 times:
  Do together:
    Lower the right leg
    Raise the left leg
  End Do together
End Loop
Lower the right leg

Modified Algorithm:
Call rightLegUp(+1)_method
Loop 10 times:
  Do together:
    call rightLegUp(-1)_method
    call leftLegUp (+1)_method
  End Do together
End Loop
Call rightLegUp(-1)_method
Stepwise Refinement

- The algorithm is not necessarily incorrect, it just needs more detail.
- Development and use of new methods means that code can sometimes be reused.
- Development and use of new methods reduces the amount of code written.

Passing Arguments

- Methods are written to accept arguments.
- Arguments are values that are passed.
- Example of arguments:
  - Direction, distance traveled.

Passing Arguments

- Passing arguments requires the use of parameters.
- Parameters are variables that hold an argument that will be passed (or used) into the method.
- Once created, the argument is required anytime the method is called.

Functions return a value where methods don’t. For the following identify inputs and possibly outputs:

- Solo for beetle
- Wheely moving around and turning near edge
- Checking if wheelies have crashed
- Making an object levitate
- Making all objects disappear

Using the class parameters

- It “real” languages, you are able to specify the type of the parameter. That allows you to use methods that all objects don’t necessarily have.
- Not being able to do that is limiting.

Using Class-Level Variables as Properties

- Properties can be added to an object through the creation of class-level variables.
- When the object is saved as a new class, the variables are saved with it.
- Common properties are:
  - color
  - opacity
  - isShowing
Using Class-Level Variables as Properties

- **Property** is a variable that belongs to an object.

- Properties hold data about that object and are referred to as **class-level variables**.

Writing Class-Level Functions

- Function is a method that returns a value back to the instruction — such as: `distance` to

- Additional data is required by the function to return back a value — for example...ExerciseGirl can runInPlace forever, but she'll get tired — a function can be created...but it needs to know how many repetitions it will take before she gets tired AND how many repetitions she has already completed.

World-Level Methods and Variables

- Rather than creating one long method...several smaller world-level methods can be created.

- Each smaller method performs a specific part of the overall animation.

- Can also create world-level variables — These variables are available to all methods in the world.

World-Level Methods

- Class-level methods give unique behaviors to an object.
- **World-level methods** are methods created for the object `world`.
- Worlds can have many methods.
- Commonly used to break a large complex algorithm into small manageable pieces — This is the **divide and conquer** approach.

World-Level Methods Example
World-Level Variables

- Variable that belongs to the world object
- World-level variables act as properties for the world
- Exist as long as the world is running (playing)
- Available to all methods in the world

World-Level Variables

- Variable that belongs to the world object
- World-level variables act as properties for the world
- Exist as long as the world is running (playing)
- Available to all methods in the world

Using Clipboards

- Clipboards are locations for storing a copy of something
- Use the clipboard in Alice to copy an instruction from one method to another
  - Drag the instruction tile to the clipboard
  - Grab the clipboard and drag back to another method
  - Very flakey – doesn’t work well with parameters

Tips for Visual Effects and Animation

- Billboards
  - Graphical images that have been inserted into the world
  - Can insert images that are JPEG, GIF, TIF
  - Known as “Billboard”
  - Images are flat, 2D with height and width no depth
  - Can be used as backgrounds or scenery or used to give instructions to the viewer!

Tips for Visual Effects and Animation

- Billboards
  - Graphical images that have been inserted into the world
  - Can insert images that are JPEG, GIF, TIF
  - Known as “Billboard”
  - Images are flat, 2D with height and width no depth
  - Can be used as backgrounds or scenery or used to give instructions to the viewer!

Fog

- Alice has a look of “mist” to the world
- The entire world becomes less visible
Tips for Visual Effects and Animation

Once an object has been put into a position or pose, the pose can be restored during an animation.

- Pose is a property of an object.
- setPose is a method of an object.

Tips for Visual Effects and Animation

Creating Dummy Objects

- Dummy objects are invisible objects that are placed in the world.
- Camera then moves to the invisible object to get different perspectives.

Tips for Visual Effects and Animation

Programming the Camera

- Camera can be programmed just like other objects.
- All the same primitive methods are available for it:
  - Point at, Move, Turn, etc.
- Camera can also be a vehicle for other objects!