Scores

- On average people dropped 5 points from the first test. Since I counted this as 5 points less, the percents are very close.
- However, some of you made huge increases or decreases in your scores.
- I think we need to offer a third (optional) test which replaces the lower of the two scores.

Optional final

- 90 points possible. Comprehensive. Given during final exam period.
- Will make it up to be a regular one hour exam – which you have an hour and a half to complete.
- (will take place after “project quiz” on final day)
- Replaces the lower of the two exam scores (so if you do worse, it hurts you)
- Must notify me (by email) by April 24th if you intend to take it.
- Most of you should NOT take the optional final.

Success

- Eagle grades are “fairly” close. The cutoffs may lower a tiny bit (say a half of a percent).
- All that is left now is assignments, participation points, and the final project.
- It is possible to be “practically perfect” on the programs – so that is a great way to improve your standing. You might do a little bit extra – as that may make up for some small mistake elsewhere.
- Don’t neglect participation points (surveys and questions) as they are weighted at 5% of the total points – which is a big swing.
Chapter 11 Objectives

- To understand:
  - what the many variables are and when and how they are used
  - what three things are specified when creating a variable
  - how a function differs from a method
  - where an object’s functions are listed
  - what primitive functions are
  - what operators are and how are they used
  - how math expressions are used in programming
  - what a collision is and ways to avoid them
  - what a string and string concatenation are

Example

- Homework for Friday, Beetle band.
- Each Beetle, plays a solo. Jumps up and down twice its height.
- Can copy instructions and substitute names.
- How can we make it easier – more reusable?

Variables

- Programs may need to store data when running and it stores that data in a variable
- A variable is a named storage location in the computer’s memory
- Scope: the area where the variable is known

Variable Types – like the “jocks” in your high school (what they look like and what they do)

- Local variables
  - belongs to a specific method
  - used only by the instructions that the variable belongs to
  - when a method stops, the variables cease to exist
- World-level variables
  - variable that belongs to the world

Creating Local Variables

- Local variables belong to a specific method (such as world.my first method)
- Variable declarations require 3 things:
  - name
  - type
  - initial value

Names and Types and Initial Values

- Variable Names
  - Must be unique within the method
  - Should be meaningful and reflect the variable’s purpose
  - Named using camelCase
- Variable Types
  - Numbers, Boolean, Objects, Other (such as String, Color, or Sound)
  - Numbers normally aren’t stored as numbers unless you will do comparisons/math with them
- Initial Value
  - Value initially stored in the variable
What are the types of the following:

- Whether you read the chapter or not
- Your library card number
- Your phone number
- The height of the tallest person
- The tallest person
- The name of the tallest person
- If you are awake
- Your grade in the class

Variables

- Tile appearance shows the type of variable – note the square before the variable name.

Creating the variable is called *variable declaration*.

Variable Assignment

- Variables have an initial value to protect against errors
- Initial value held until a different value is assigned
- New values can be assigned while the method is running
  - *Set instructions* can be created to store (or set) different values
  - Request to set a value occurs when variable tile dropped into Methods Editor. “Set value” is the first choice.
  - Sometimes you have to put in a dummy value to be replaced later.

Functions

- Like a method – but returns a value. Example, give me the area of a figure.
- Each object has a set of primitive functions
- List of functions is categorized
  - Boolean logic (true/false)
  - Math
  - Random
  - String
  - Ask user
  - Mouse (distance from edge)
  - Time
  - Advanced math
  - Other

Snowman who jumps up and down his height. Other variables?

- World has three primitive functions that ask for user input
- Each function is displayed in a dialog box

Asking the User for Input
Primitive Object Functions

- Each object has primitive functions different from the world’s primitive functions
  - Proximity
  - Size
  - Spatial relation (left of, right of, …)
  - Point of view
  - Use “print” (from bottom of screen) to see what is returned.

Creating Math Expressions

- Math Operators are used to create math expressions
- Math expressions perform a calculation and return a value
- Operands appear on both sides of an operand
  - “payRate” and “hoursWorked” are operands
- This interface is a bit awkward. “Just typing” would be much easier – but error prone.
- If you try to drag an illegal operand, it just ignores you.

Sample Problem

- We want the horse and rider to jump over the fence.
- See horseJump.a2w
- Suppose we want the horse to jump a random height and fall down if it doesn’t jump high enough.
  - Notice “as seen by” after horse has fallen.
  - Notice if/else statement

Working with Strings and Text

- A string is a sequence of characters
- Strings represent:
  - names
  - addresses
  - messages
  - etc.

Asking the User to Enter a String

- World’s primitive function can ask user to enter string (text)
- The function returns the text the user enters
- Joining separate sets of strings is called concatenation
  - “Hello” joins with the user’s name

Joining Strings

- String Concatenation
  - Joins string a with string b (see world functions)
  - Strings a and b are arguments
  - Does not change the values of strings a or b…but combines the two strings
  - Strings are NOT just text…but any alphanumeric characters
- Can convert numbers to a string for use as well
At seats, prompt user for name

When you "ask user" for a string, system will want a prompt. In this case, the prompt was "Enter a name:"

```python
world.concert()
first = <None>, name = Hello, greeting = Hello good buddy

print(lennonBeetle's position)

name set value to (ask user for a string question = Enter a name:)

name set value to (greeting joined with name)
lennonBeetle say name
```

Text can be added to Alice worlds

3D text option is available at the end of the local gallery