Review

- Assignment #3 Clarifications
- Parts of a class statement
- Class and Instance Types
- Modeling objects with classes
- The __str__() method
- Assignment vs. Copy vs. Deep Copy

Exam 1 Topics

- Coordinate System
- Colors
  - red, green, blue, alpha, 0..255
- Drawing Primitives
  - line(), rect(), ellipse(), triangle(), quad(), …
  - background(), text(), …
- Formatting Primitives
  - fill(), stroke(), …
- Sketch
  - window(), width(), height(), mouseX(), mouseY(), …
- Random Number Generation
  - How to specify a range

Exam 1 Topics

- Identifiers (variables)
- ‘None’ keyword
- Types
  - int, float, str, bool, list, classobj, instance, …
- Operators
  - (next slide)
- Expressions
  - Mathematical, Logical, Relational
  - How to write, use and evaluate
- Visibility, Lifetime
  - ‘global’ statement

Operators

Mathematical (+, -, *, /, **, //, %, ...)

- Perform standard mathematical operations.
- PEMDAS

Relational (<, >, ==, !=, <=, >=, ...)

- Test relationship between related expressions.
- Always returns a boolean value (True or False).

Logical (and, or, not)

- Logical conjunction (and), disjunction (or), negation (not).
- Always return a boolean value (True or False).

Exam 1 Topics

- Functions
  - How to call and existing function
  - How to define a new one, pass arguments, return a value
  - ‘def’ keyword, ‘:’, whitespace (tabs)
- Events
  - Handling (+=, -=), Raising
  - onMousePressed, onLoop, …
- Conditionals
  - if, elif, else
  - How to define, use, trace
- Loops
  - while, for-in
  - How to define, use, trace

Exam 1 Topics

- Lists
  - define, use, modify
  - Looping over items, appending, removing, …
- Transformations
  - translate(), rotate(), scale()
  - pushMatrix(), popMatrix()
- Classes and Objects
  - Define, use
  - Instance variables
  - Methods
  - Dot-notation

Exam 1 Topics

- Problem Solving
  - Write expressions to test a condition
  - Formulate conditionals to direct program execution
  - Using global variables to maintain state
  - Modify a complex drawing with transformations
  - Using lists to maintain state of multiple objects
  - Define functions to perform reusable actions
  - Define a class to model an object
  - Initialize instance variables to hold object state
  - Define methods to implement object behavior
  - …