Lecture 7 Announcements

• Please check the end of the Lecture 6 slides (slides 25-29) for many announcements:
  http://www.cs.cornell.edu/courses/cs1110/2017sp/lectures/02-14-17/presentation-06.pdf

• Incorrect link for how to break up long lines in Section 10 of Assignment 1. Watch course website for announcements about A1:
  http://www.cs.cornell.edu/courses/cs1110/2017sp/announcements.php

Type: Set of values and the operations on them

• Want a point in 3D space
  * We need three variables
  * x, y, z coordinates
  * What if have a lot of points?
    * Vars x0, y0, z0 for first point
    * Vars x1, y1, z1 for next point
    * …
    * This can get really messy
  * How about a single variable that represents a point?

Objects: Organizing Data in Folders

• An object is like a manila folder
• It contains other variables
  * Variables are called attributes
  * These values can change
• It has an ID that identifies it
  * Unique number assigned by Python
    (just like a NetID for a Cornellian)
  * Cannot ever change
  * Has no meaning; only identifies

Classes: Types for Objects

• Values must have a type
  * An object is a value
  * Object type is a class
• Modules provide classes
  * Will show how later
• Example: geom
  * Classes: Point2, Point3

Classes: Types for Objects

• Classes are how we add new types to Python

Constructor: Function to make Objects

• How do we create objects?
  * Constructor Function:
    * Same name as the class
    * Example: Point3(0,0,0)
    * Makes an object (manila folder)
    * Returns folder ID as value
  * Example: p = Point3(0, 0, 0)
    * Creates a Point object
    * Stores object’s ID in p
    * You need the assignment to p to be able to use the object later
Accessing Attributes

- Attributes are variables that live inside of objects
  - Can use in expressions
  - Can assign values to them
- Access: `<variable>..<attr>`
  - Example: `p.x`
  - Look like module variables
- Putting it all together
  - `p = geom.Point3(1,2,3)`
  - `p.x = p.y + p.z`

Object Variables

- Variable stores object name
  - Reference to the object
  - Reason for folder analogy
- Assignment uses object contents
  - Example: `q = p`
  - Takes contents from `p`
  - Puts the contents in `q`
  - Does not make new folder!
- This is the cause of many mistakes in this course

Exercise: Attribute Assignment

- Recall, `q` gets name in `p`
  ```python
  >>> p = geom.Point3(0,0,0)
  >>> q = p
  ```
- Execute the assignments:
  ```python
  >>> p.x = 5.6
  >>> q.x = 7.4
  ```
- What is value of `p.x`?
  
  A: 5.6  
  B: 7.4  
  C: `id4`  
  D: I don’t know

Call Frames and Objects

- Mutable objects can be altered in a function call
  - Object vars hold names!
  - Folder accessed by both global var & parameter
- Example:
  ```python
  def incr_x(q):
      q.x = q.x + 1
  >>> p = geom.Point3()
  >>> incr_x(p)
  ```