Object Memory Model

Oliver W. Layton

CS151: Computational Thinking: Visual Media

Lecture 11, Fall 2020

Friday September 18
Project 4

Make a race track scene and have two turtles race each other around a track.

• Shape of race course is up to you. Circle is the simplest/recommended.

• One Turtle object with dedicated job of drawing the background scene (race track, audience, etc.)

• Two different Turtle objects are the racers.

• Two other Turtle objects display their score ("scorekeepers") — how many laps Turtle 1 and Turtle 2 complete.

• All drawing should use objects! No code that looks like `turtle.function()`
Project 4 organization

- **object_shapelib.py**: Like `better_shapelib.py`. All your shape and scene drawing functions. One `Turtle` object created in `object_shapelib.py` takes care of all this drawing.
  - Running `object_shapelib.py` should draw your race track scene.

- **race.py**: Make the `Screen` object, 2 racing turtles, 2 score keeper turtles. Moves turtles around the race course.
  - Running `race.py` should draw your race track scene, have the turtles race and display their current score.
Symbol table with objects

- So far, we know that modules and functions have their own symbol tables.

- Basic data types do not have their own symbol tables.

- **Important new concept:** Objects have their own symbol tables.

- Let's look at code similar to lab/project with objects, then write out the symbol table.