Objects

Oliver W. Layton

CS151: Computational Thinking: Visual Media

Lecture 10, Spring 2021

Wednesday March 3
Objects

• New type of data (other than string, float, int, boolean).

• Two key ideas

  1. Objects help us organize heterogenous data/information related to a "thing" we care about. **Objects "know stuff".**

  2. Objects allow you to perform operations (actions) and manipulate the stored data. **Objects can "do stuff".**
1) Objects group data (1/2)

Maintaining information about cars:

car1_color = 'blue'
car1_tankSize = 11
car2_color = 'silver'
car2_tankSize = 15
car3_color = 'black'
car3_tankSize = 9

The data are not *really* associated with car1, car2, and car3...they are just names we came up with.
1) Objects group data (2/2)

car1_color = 'blue'
car1_tankSize = 11
car2_color = 'silver'
car2_tankSize = 15
car3_color = 'black'
car3_tankSize = 9

Objects *actually* group together data related to an *instance* of a *common* thing.

Common thing here: Cars.

3 cars → 3 unique Car objects ("3 instances of a car"): Car 1, Car 2, Car 3 — each only stores data relevant to *itself*. Let's draw unique symbol tables.
2) We perform actions on objects to change their data

Example: Car 1 gets painted gray. How do we tell the object that its existing color (blue) needs to change?

We call a function on it, passing in the new color (blue):

car1.setColor('gray')

• How do find out what Car 1 and Car 2's colors currently are? We call another function on each object that returns the current color string:

color1 = car1.getColor()
color2 = car2.getColor()
print(color1)  # gray
print(color2)  # silver
Summary of new ideas

• Each object holds and groups together info that makes that object **unique**.

• Call functions on the objects to do actions like set and get their unique data (`car1.getColor()` vs `car2.getColor()`).

• Functions called on objects (`object.function()` called **methods**).

• The unique object GOES BEFORE THE DOT (`car1.function()` vs. `car2.function()`).
turtle with objects

Enables us to do cool things that weren't possible before!

Two "things" / types of objects we will work with:

1. Turtle: You can now create more than one turtle that draws on the screen! Each is an object and has unique properties (color, shape, position on canvas, pen width, etc.)

2. Screen: represents the canvas (pop-up window in which turtle draws stuff). Controls canvas size, background color, etc.
How do we create objects in the first place?

• Call a special method called the **constructor**. It returns a new object.

• Convention: constructor starts with a capital letter.

• To create two *Turtle* objects to draw stuff (later):

```python
turt1 = turtle.Turtle()
turt2 = turtle.Turtle()
```

• The two objects are **independent**: changing `turt1`'s color will not affect `turt2`. 
Let's write some turtle code with Turtle and Screen objects