

CS 151 Lab 11: oops!

Overview

We'll be focusing on object-oriented programming (OOP) in this lab. You'll start by looking at a buggy program to review the basics of OOP and the anatomy of a class in Python. Then you'll define a simple class from scratch!

Learning outcomes

Practice working with OOP:

- Understanding the anatomy of a class definition (instance properties and instance methods).
- Implementing a design for a class.

Practice programming skills:

- Debugging in Python.
- OOP syntax in Python.

Submission

1. Submit your group's **marker.py** and **main.py** program files on Gradescope; be sure to have all group members on the submission, but you only need one submission.
2. Using Week 11's forum, post a reflection about your experience:
 - *How was today's lab?*
 - *Did you find your comfort level shift for:*
 - *Debugging*
 - *OOP*
 - *Any other observations?*

Required tools

To complete this lab, you will use:

- A web browser (e.g., Firefox, Chrome)
- VS Code

Part 1: find the bugs in the flowers

Let's review the basics of OOP using this [worksheet](#).

Part 2: markers

In your groups, work to implement a class that represents a Marker whose cap color can change. You'll be following the design below.

Marker.py

Define a simple class that represents a marker. Write your class definition in a file named **Marker.py**, adhering to the following specification:

- The class should be named **Marker**.
- There should be a two instance properties of type **str** that maintain:
 - the color of the marker
 - the color of the marker's cap
- The constructor should take in a single argument for the color of the marker. The cap should be initialized to the same color.
- There should be *getters* for each instance property called **getColor** and **getCapColor**.
- There should be a *setter* to change the marker cap color (because this happens in real life!) called **setCapColor**.
- Since markers can't change color, there will be *no setter for the marker's color*.
- There should be an instance method called **__str__** that takes in no parameters and returns a string representation of the marker. If the marker is red with a blue cap, for example, it should return "A red marker with a blue cap."

main.py

In a file **main.py**, define a main function that:

- Instantiates a new Marker instance that represents a red marker and assigns it to a variable called redMarker.
- Instantiates a new Marker instance that represents a green marker and assigns it to a variable called greenMarker.
- Prints information about the greenMarker.
- Prints information about the redMarker.
- Sets the color of the cap of the red marker to be blue.
- Prints information about the redMarker.

Remember: to use the class Marker which is defined in the file Marker.py, include the following line in main.py
from Marker import Marker

Submission checklist

- Gradescope
 - Submit your group's **Marker.py** and **main.py** program files on Gradescope; be sure to have all group members on the submission, but you only need one submission.
- Moodle:
 - Use Week 11's forum, post a reflection about your experience
 - *How was today's lab?*
 - *Did you find your comfort level shift for:*
 - *Debugging*
 - *OOP*
 - *Any other observations?*