

CS 106

Introduction to Computer Science I

07 / 22 / 2026

Instructor: Michael Eckmann

Today's Topics

- Questions / comments?
- More Object Oriented programming in Python
 - Review
- Understanding how = works on lists/objects vs.
- Write cut method in Deck
- Public vs. private instance variables

Modules, classes

- Review
- Module
 - A python file containing code, defs, classes
- Class
 - Instance variables
 - Methods
 - Constructor method
- Objects

classes

- Class (general description of the data and behaviors of something)
 - Instance variables (the data)
 - Methods (the behaviors)
 - Constructor method
- Objects
 - Objects are instances of a class type

classes

- Instance variables (the data)
 - Specified by assigning to something with self. before it
 - e.g.

```
self.rank = r # this line in Card
# creates an instance variable named
# self.rank
```
 - Other variables within methods of a class are local variables to those methods ---
 - e.g.

```
r and s in the constructor
rank_str in __str__ method
```

classes

- Constructor method
 - named `__init__`
 - has `self` as first parameter
 - job is to create and assign values to instance variables
 - is called when creating an object of the class
 - e.g.

```
mycard = playing_cards.Card(5,0)
# line above calls __init__
# (which is the constructor) in Card class
```

classes

- Other methods within a class
 - Must have self as first parameter to be able to access the instance variables
 - e.g. see all the methods (shuffle, deal, etc.) in Deck

An aside: Copying vs. assigning lists

- Suppose you have a variable whose value is a list like:
`my_list = [1,6,3,42]`
- If you assign `my_list` to another variable like:
`list2 = my_list`
- Both `list2` and `my_list` are referring to the SAME list.
- So if we changed an element of `list2` like so:
`list2[0] = 99`
- It affects the only list in existence --- both `list2` and `my_list` will now refer to: `[99,6,3,42]`
- Contrast this with a copy

Deck

- Let's write cut method in Deck

Public vs. private instance variables

- Instance variables that start with `__` (two underscores) are private. Instance variables that start with no underscores are public.
- I'll show the effect of private / public instance variables.