

CS 106

Introduction to Computer Science I

07 / 21 / 2026

Instructor: Michael Eckmann

Today's Topics

- Questions / comments?
- Object-oriented programming in Python
 - classes

class

- Recall what a type is in Python. e.g. str is a type for string data, int is a type for integer data and so on.
- Python allows programmers to define their own type by creating a class.
- For example:
 - the str class (which is built in to Python) contains data and methods that work on that data.

class

- Think of a **class** as containing
 - Data (aka **instance variables**)
 - **Methods** (functions that work on one or more instance variables)
- An **object** is an instance of a class.
- Example:
word = 'Skidmore'
another_word = 'College'
- word and another_word are both objects (aka instances) of the class str.

class

- Consider a class named Car
- Examples of objects of type Car:
 - a Blue Subaru Forester VIN# 3495235664
 - a Grey Chevy Spark VIN# 43255543243
- Any other examples you can think of for classes and objects of that class?

class

- To allow a class to have objects created of that class type (that is to be an instantiable class) a special method called the constructor should be defined.

```
def __init__(self)
```

- Can have more parameters but self should be first.
- Instance variables are preceded by **self**. within methods of the class.
- The `__init__` method is called automatically when instantiating an object of a class type.
- e.g. `__init__` in str class is called when we do:

```
word = 'Skidmore'
```

- Or more explicitly:

```
another_word = str('College')
```

class

- 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.

class

- Let's define two classes (Card and Deck) in a module named `playing_cards`.
- Let's put code for a program in a different module (python file) that creates a deck of cards and does something with them.