Intro to Object-Oriented Programming (OOP)

Extra Credit Opportunity

- If you attend 3 tutoring sessions before Quiz 03 (April 17) AND you score below 60% on the quiz, we will add 5 percentage points to your Quiz 03 grade.
- If you attend 8 tutoring sessions total this semester AND you score lower than 70% on the final exam, we will add 5 percentage points to your final exam grade.

See Tutoring details on the site's Support page!

Reminders

- **Quiz 03** on April 17
- **Final exam** on Thursday, April 30th at 4pm (Common Hour)
  - Have another exam at this time, or 3 exams in 24 hours? Please complete:
    - UNC's Official Final Exam Excuse Form
    - Our internal form for tracking requests
  - **Makeup exam** on Friday, May 1 at 12pm in Sitterson Hall (SN) 014

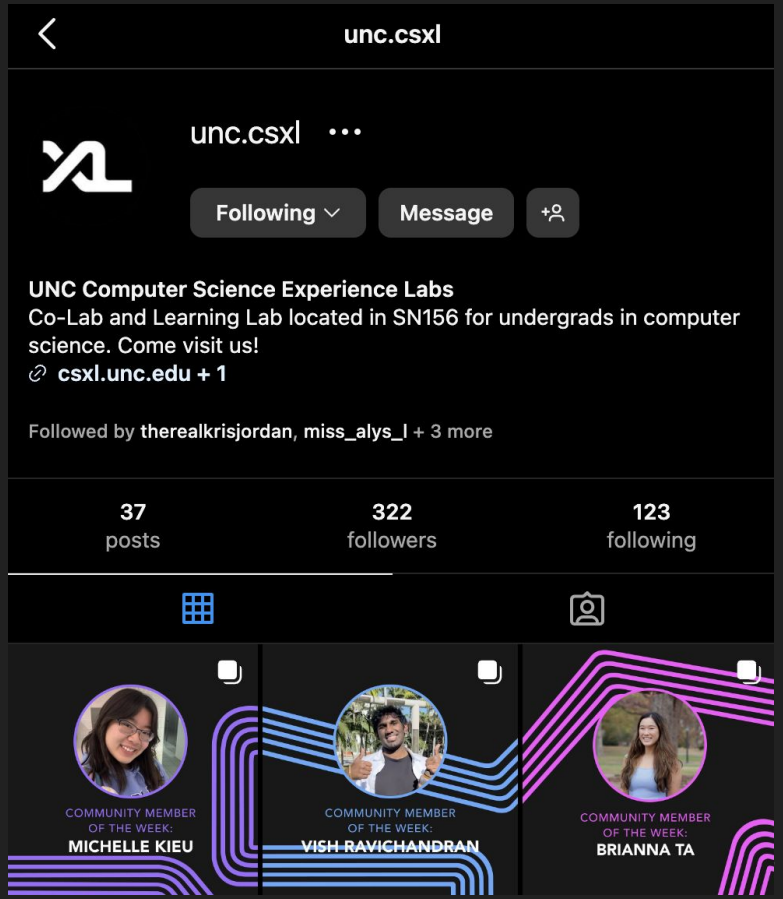
Modeling an Instagram profile with code

What data should we keep track of?

```
username: str = "unc.csxl"
bio: str = "UNC CS Experience Labs"
posts: int = 37
followers: int = 322
following: int = 123
private: bool = False
```

What behaviors would be useful?

- View # followers or following
- Write or update a bio
- (Un)follow an account
- Make an account private/public



Modeling an Instagram profile with code

What data should we keep track of?

```
username: str = "unc.csxl"
bio: str = "UNC CS Experience Labs"
posts: int = 37
followers: int = 322
following: int = 123
private: bool = False
```

What behaviors would be useful?

- View # followers or following
- Write or update a bio
- (Un)follow an account
- Make an account private/public

Instagram has over **2 billion** user profiles…

What challenges could we encounter?

It’d be nice to be able to bundle these attributes and functions into one object per profile…

What are objects *in the real world*?

What are objects *in the real world*?

Things that can be perceived, used, or interacted with

They can be *physical*:

- Chair is a type of furniture
- Human is a type of mammal
- Fork is a type of utensil

or *abstract*:

- Lecture is a type of event
- Friendship is a type of relationship
- Learning is a type of experience

And they all serve distinct purposes!

What are objects *in Python*?

Many types of data in Python:

```
23      "hello world!"        3.14159        [24, 26, 25, 27]

{110.001: "Hinks", 110.002: "Hinks", 110.001: "Lytle"}   True
```

Every object has:

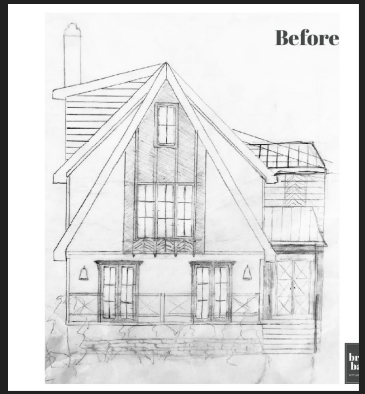
- A type
- An internal data representation
- A set of procedures to interact with the object

An object is an instance of a type

- `23` is an instance of an `int`
- `"hello world!"` is an instance of a `str`

Classes and **objects**

- Think of a **class** as a blueprint/ template
  - Defines attributes and behaviors its objects will have
- An **object** is an *instance* of a class
  - E.g., if the class is the blueprint, the object is the house!
  - Has all the specified attributes and behaviors
  - Different objects share these attributes and behaviors, but are distinct!



Modeling an Instagram profile with a `class`

declaring a new data type!

```
class Profile:
```

Modeling an Instagram profile with a `class`

declaring a new data type!

```
class Profile:
    username: str
    bio: str
    followers: int
    following: int
    private: bool
```

declaring attributes
(every Instagram profile has these!)

Modeling an Instagram profile with a `class`

declaring a new data type!

```
class Profile:
    username: str
    bio: str
    followers: int
    following: int
    private: bool

    def __init__(self):
        self.username = "usr9"
        self.bio = ""
        self.followers = 0
        self.following = 0
        self.private = False
```

declaring attributes
(every Instagram profile has these!)

initializing attributes
(what are the default values?)

Modeling an Instagram profile with a `class`

declaring a new data type!

```
class Profile:
    username: str
    bio: str
    followers: int
    following: int
    private: bool

    def __init__(self):
        self.username = "usr9"
        self.bio = ""
        self.followers = 0
        self.following = 0
        self.private = False

my_prof: Profile = Profile()
```

declaring attributes
(every Instagram profile has these!)

initializing attributes
(what are the default values?)

Construct (instantiate) a new profile!

Modeling an Instagram profile with a `class`

declaring a new data type!

```
class Profile:
    username: str
    bio: str
    followers: int
    following: int
    private: bool

    def __init__(self):
        self.username = "usr9"
        self.bio = ""
        self.followers = 0
        self.following = 0
        self.private = False

my_prof: Profile = Profile()
my_prof.username = "comp110fan"
print(my_prof.private)
```

declaring attributes
(every Instagram profile has these!)

initializing attributes
(what are the default values?)

Memory diagram

```
1 class Profile:
2    username: str
3    bio: str
4    followers: int
5    following: int
6    private: bool
7
8    def __init__(self):
9        self.username = ""
10       self.bio = ""
11       self.followers = 0
12       self.following = 0
13       self.private = False
14
15
16 my_prof: Profile = Profile()
17 your_prof: Profile = Profile()
18 your_prof.username = "unccompsci"
19 my_prof.username = "unc.csxl"
20
21 print(my_prof.username)
```

Returning to our goal: modeling an Instagram profile with code

What data should we keep track of?

```
username: str = "unc.csxl"
bio: str = "UNC CS Experience Labs"
posts: int = 37
followers: int = 322
following: int = 123
private: bool = False
```

What behaviors would be useful?

- View # followers or following
- Write or update a bio
- (Un)follow an account
- Make an account private/public

How can we write code to enable these actions for any Instagram account?