COMP110

Intro to Object-Oriented Programming (OOP)

Reminders

- EX05 (Dictionary Unit Tests) due Sunday at 11:59pm
  - Want help? Visit Office Hours!
    - 11am–5pm today
    - 1–5pm on Sunday

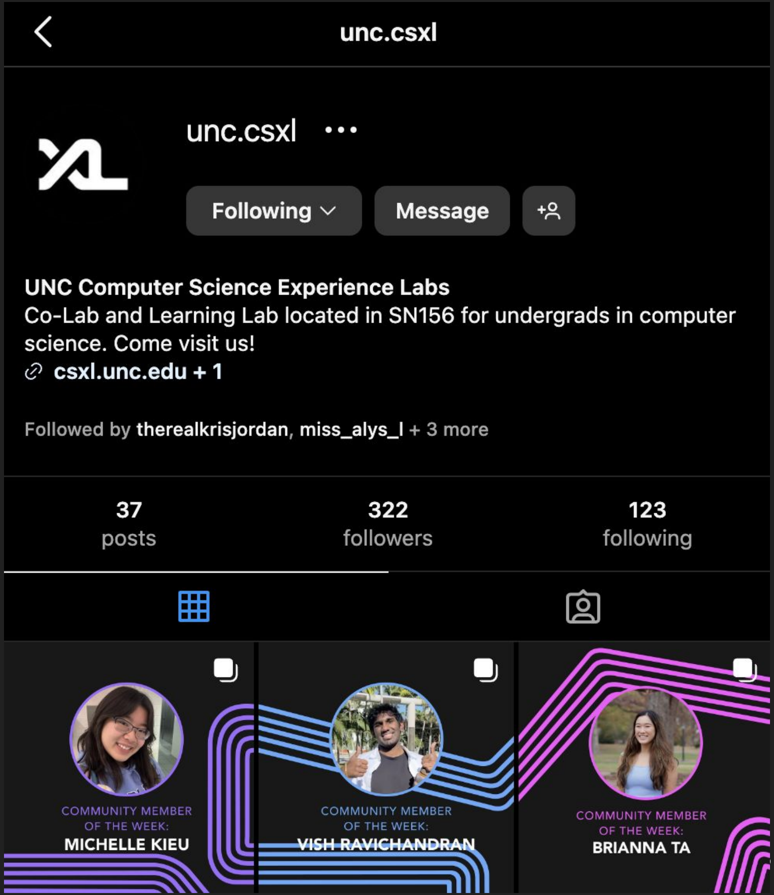
Modeling an Instagram profile with code

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


unc.csxl
unc.csxl
Following
Message

UNC Computer Science Experience Labs
Co-Lab and Learning Lab located in SN156 for undergrads in computer science. Come visit us!
csxl.unc.edu + 1

Followed by therealkrisjordan, miss_alys_l + 3 more

37 posts · 322 followers · 123 following

COMMUNITY MEMBER OF THE WEEK: MICHELLE KIEU
COMMUNITY MEMBER OF THE WEEK: VISH RAVICHANDRAN
COMMUNITY MEMBER OF THE WEEK: BRIANNA TA

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:

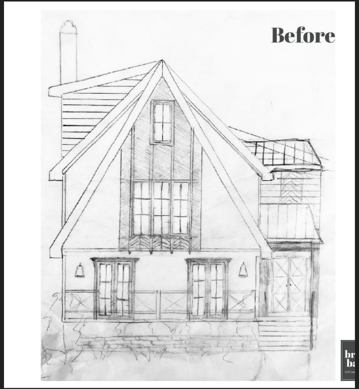
- A type
- An internal data representation
- A set of procedures to interact with the object

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!



After
(Virtual Image
385

After
(Virtual Image
385

Modeling an Instagram profile with a `class`

$\overbrace{\texttt{class Profile:}}^{\text{declaring a new data type!}}$

```python
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!

```python
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!

```python
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

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

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


my_prof: Profile = Profile()
your_prof: Profile = Profile()
your_prof.username = "unccompsci"
my_prof.username = "unc.csxl"

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?