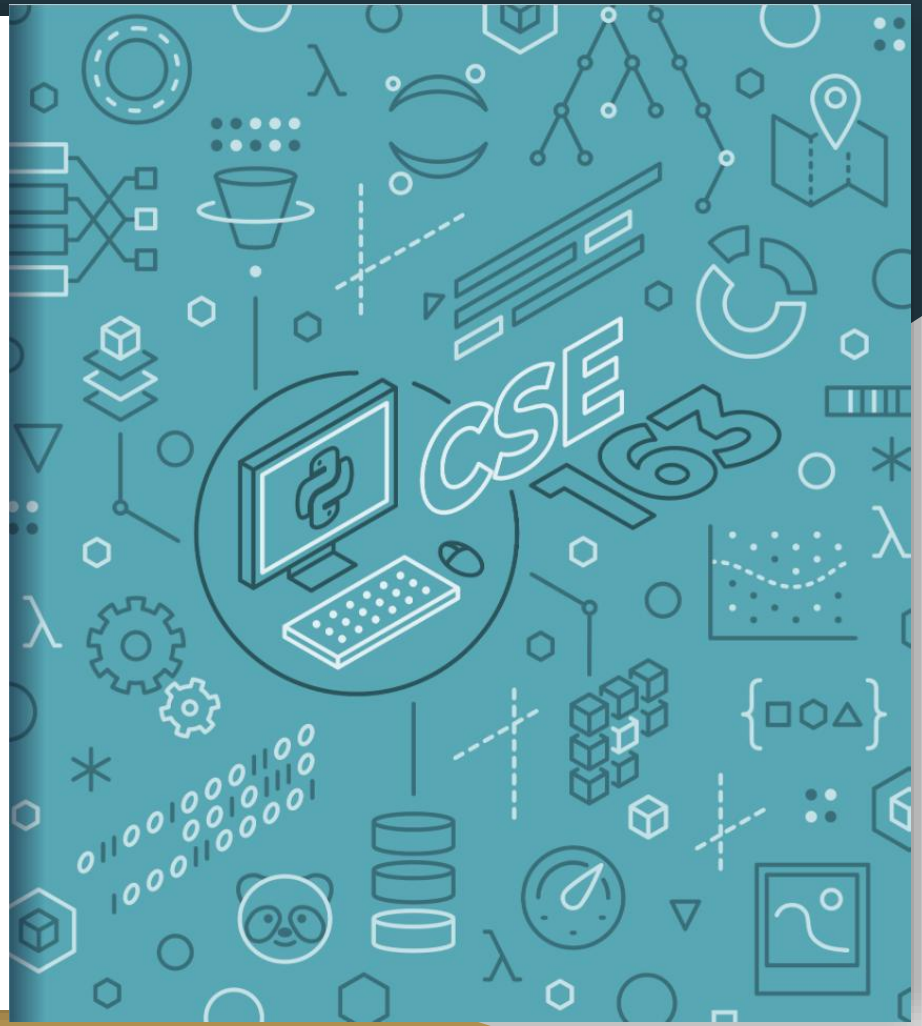
CSE 163
Inheritance

Adrian Salguero
Spring 2026

💭 Icebreaker (discuss with neighbors): ‘
What is your favorite food spot on the Ave?
Add to our Slido!

slido.com
#cse163



Announcements

- **Take Home Assessment 2: Networks** due Thursday, April 23rd at 11:59pm!
  - Attend office hours if you are stuck or need help
- **Lesson 7 Canvas Quiz** due tonight at 11:59pm!
- **Checkpoint 2** due Monday, April 20th at 11:59pm!
- **THA 1 Peer Reviews** due on tonight at 11:59pm!
  - **Instructions**
  - "I Like", "I Wish", and "What If"
  - You are only ***assessed by the review you give***
  - If you are assigned a ***blank notebook***, please write that it was blank on your response

Lambdas

- Lambdas allow us to define functions within the body of our code!

```
def get_dog_name(d):
    return d.get_name()

sorted(dogs, key=get_dog_name)
```

```
sorted(dogs, key=lambda d: d.get_name())
```

What do we have to write this pattern?

```
def main():
    print('Hello world')

if __name__ == '__main__':
    main()
```

- If you don’t write this pattern, your main method will run if you import the file!
- What are some potential downsides of this happening?

Special Methods

Syntax	Method
`x < y`	`x.__lt__(y)`
`x > y`	`x.__gt__(y)`
`x == y`	`x.__eq__(y)`
`x <= y`	`x.__le__(y)`
`x >= y`	`x.__ge__(y)`
`print(x)`	`print(x.__str__())`
`x[i]`	`x.__getitem__(i)`
`x[i] = v`	`x.__setitem__(i, v)`

Inheritance

```
class Dog:
  def __init__(self, name: str) -> None:
    self._name: str = name


  def bark(self) -> None:
    print(self._name + ': Woof')
```

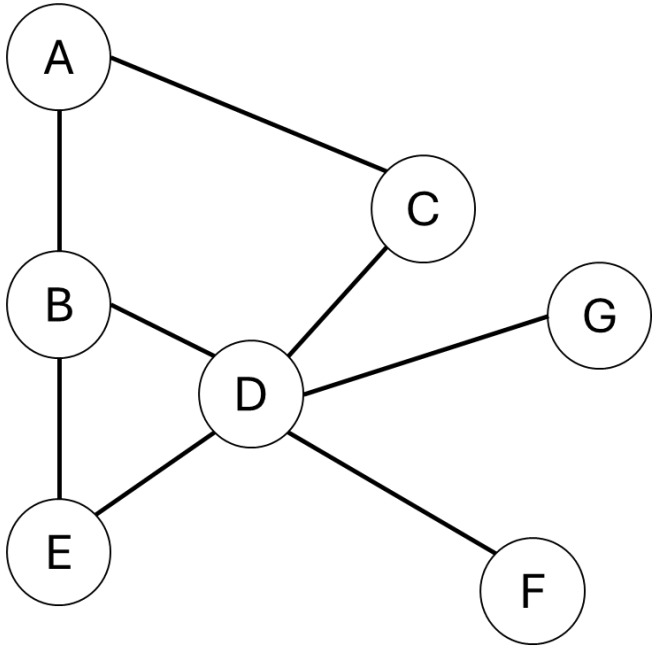
```
class ServiceDog(Dog):
  def __init__(self, name: str, service: str) -> None:
    super().__init__(name)

    self._service: str = service


  def bark(self) -> None:
    print(self._name + ': *alert bark*')
```

Graphs

- Nodes + Edges
  - Nodes represent data in the graph
  - Edges represent relationships between the data nodes
- Suppose the diagram on the right represents a “social network”
  - B’s friends: A, D, E
  - C’s friends: A, D
  - B + C’s mutual friends: A, D
- What else can graphs represent?



THA 2 Tips

- Start early!
- `graphs.py`
  - Edges and nodes are in a list for convenience
- `analysis.py`
  - Read through the start code thoroughly (come to office hours if you have questions)
  - Consider what different data structures represent
  - The two recommendation algorithms will be fairly similar
- Creative Component
  - Refer to the `(node, dict)` syntax for adding node attributes

Group Work: Best Practices

- When working with a new group for the first time:
  - Introduce yourself!
  - If possible, angle one of your screens so that everyone can see and discuss together
  - Be respectful of each other and allow everyone to speak

- Tips:
  - Start with making sure that everyone agrees to work on the same problem
  - Allow everyone to contribute or a chance to ask questions.
  - Ask if everyone agrees and periodically ask each other questions.
  - Call TAs or Adrian over if you need any help.
  - Don’t sit in silence.