



# Protocols and Composition

**COMP110 - CL21**

2024/04/23

# Closing Out the Semester

- EXo7 - Compstagram - Build photo filters with classes, objects, and algorithms
  - Out tonight and due LDOC
- Final Exam
  - Friday, May 3rd at 8am
  - Conflict? Makeup: Saturday, May 4th at 12pm in SNo14

# Warm-up

1. In a new file, create a new lecture directory **cl21**. In it, a file named **theater.py**
2. Implement a class named **Ticket** with the following features:
3. One *attribute* named **matinee** that holds a boolean value. This attribute should be initialized by a parameter passed to a constructor (hint: `__init__`)
4. Add a `__str__` method such that its returned string representation is either "ticket" or "ticket (matinee)" depending on the value of the matinee attribute.
5. Add a **price** method that returns a float. Choose your own ticket prices for a matinee versus non-matinee ticket.
6. Above the class definition, add a **main** function that constructs two instances of Ticket objects, one matinee, one standard, and prints information about them.

# Adding Concessions: Drink and Popcorn

1. In the same file, **theater.py**, define the following classes, following the Ticket class:

2. Add a **Drink** class with:

- \* a **size** attribute initialized in a constructor
- \* a **\_\_str\_\_** method implementation that produces a reasonable string representation
- \* a **price** method that returns a price depending on the size of the drink, you choose prices

3. Add a **Popcorn** class with:

- \* **size** and **butter** attributes, both initialized in the constructor
- \* a **\_\_str\_\_** method implementation that produces a reasonable string representation
- \* a **price** method that returns a price depending on the attributes, you choose prices

4. In the main function, construct some Drink and Popcorn objects, print their string representation and prices

# **How do we model a Receipt with Ticket(s), Popcorn(s), and Drink(s)?**

## **Code-Along: Let's implement a Receipt**

# Notes on Structural Types using `typing.Protocol`

# Notes on Composition - "has a" relationship