

Hack110 Sign-Up Form!

When? Saturday, April 5th from 10 AM - 12 AM (Midnight)

Where? In Sitterson Lower Lobby

Who can join? Anyone in COMP 110! No prior experience required. Bring a partner or come as yourself (we'll have team-building activities if you want a partner)

Come for a fun day of coding, workshops and events (**food and CLE credit will be provided**):

- Choose between web development or game development track
- Go to various **workshops & events** such as: Navigating the CS Major, Resume workshop, ice cream station, and kahoot trivia and MORE!
- Link: Sign-Up Here! Or via the QR code 
- **Sign-Up form EXTENDED TO Monday, March 31st at 11:59 pm**
 - Spots are limited! So we'll prioritize interest!
 - If you have a partner, **ONLY ONE OF YOU** has to sign up - you will just enter your partner's info in the form.

Sign-Up Here!





CL25: Review!

Announcements

- *Please submit regrade requests for Quiz 02 by **this Friday at 11:59pm***
- EX03 due *today!*
- Quiz 03 on Friday
 - **Review Session tonight** at 6:15pm in Fred Brooks (FB) 009
 - Other ways to prep:
 - Review Quiz 02 to address your gaps in understanding
 - Finish EX03 and review your code – try to diagram example function calls!
 - Practice Quiz (solutions and explanation video linked on the site)
 - Please visit us in Office Hours + Tutoring!

Match the Data Structure to its Application

Set

List

Dictionary

Store a bunch of
tasks in a specific
order

Keeping track of
inventory in a store
(names of items and
the number in stock)

Store the jersey
numbers of UNC's
basketball team

Match the Data Structure to its Application

List

Store a bunch of
tasks in a specific
order



Dictionary

Keeping track of
inventory in a store
(names of items and
the number in stock)



Set

Store the jersey
numbers of UNC's
basketball team



Data structure	Allows duplicates?	Ordered?	Fast lookups?	Use Case
list []				Ordered collections
set { }				Unique values, membership testing
dictionary {key: value}	 (duplicate values allowed; keys must be unique!)	It's complicated		Mappings, fast lookups, counting

Memory Diagram

```
1  def group_names(names: list[str]) -> dict[str, int]:
2      groups: dict[str, int] = {}
3      first_letter: str
4      for n in names:
5          first_letter = n[0]
6          if first_letter in groups:
7              groups[first_letter] += 1
8          else:
9              groups[first_letter] = 1
10     return groups
11
12     ppl: list[str] = ["Karen", "Emily", "Kris"]
13     output: dict[str, int] = group_names(names=ppl)
14     print(output)
15     output["I"] = 1
16     print(output)
```