

Hack110 Sign-Up Form!

When? Saturday, November 8th 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, find one there, or go solo!

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

- Choose 1 of many tracks like web dev, game dev, etc - and WIN PRIZES too!
- Go to various **workshops & events** such as: Navigating the CS Major, Resume workshop, ice cream station, and kahoot trivia and MORE!
- Link: Sign-Up via the QR code

- **Sign-up form closes Monday, October 20th 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.

Instagram



Sign-Up Here!





Reference Types & Practice with Lists

Announcements

- **CQ02: Memory Diagram with Lists** due *Sunday, Sep 28*

```
1  """An example of primitive vs. reference types."""
2
3  a: int = 0
4  b: int = a
5  b += 1
6  print(f"a is: {a}")
7  print(f"b is: {b}")
8
9  c: list[int] = [b, 4]
10 d: list[int] = c
11
12 d.append(13)
13
14 print(c)
```

Your turn: write a function called `gen` that takes as input an int called `n`, and returns a `list` of integers from 0 up to, but not including `n`

```
1 def gen(stop: int) -> list[int]:
2     """Generate a list from 0 to stop, not inclusive."""
3     i: int = 0
4     acc: list[int] = []
5     while i < stop:
6         acc.append(i)
7         i = i + 1
8     return acc
9
10
11 print(gen(3))
```

Diagramming a Nested List (Your CQ to submit!)

```
1  def sum2d(xs: list[list[int]]) -> int:
2      """Calculate the sum of a 2-dimensional list of lists."""
3      total: int = 0
4      row_i: int = 0
5      while row_i < len(xs):
6          col_i: int = 0
7          while col_i < len(xs[row_i]):
8              total += xs[row_i][col_i]
9              col_i += 1
10         row_i += 1
11     return total
12
13
14 values: list[list[int]] = [[1, 2, 3], [3, 4, 5]]
15 print(sum2d(values))
```

```
17 def mutator(x: int, exes: list[int]) -> int:
18     """An impure function..."""
19     x += 1
20     exes[0] += 1
21     y: int = x + 1
22     print(f"mutator x: {x}, exes: {exes}, y: {y}")
23     return x
24
25
26 x: int = 0
27 exes: list[int] = [0]
28 y: int = 0
29 print(f"global before x: {x}, exes: {exes}, y: {y}")
30 y = mutator(x, exes) + 2
31 print(f"global after x: {x}, exes: {exes}, y: {y}")
```