

Recursion, Positional Arguments and Default Parameters

Today starts as a Paper + Pencil or Tablet + Pencil day... please keep laptops stowed away!

COMP110 - CL07

2024/02/08

Announcements

- EXo2 Grade Calculator - Due Wednesday 2/14
- Office Hours Closed Saturday 2/10 through Tuesday 2/13 - Well-being
- QZ01 - Thursday 2/15

Warm-up: Diagram *at least* 2 function call frames... *with Heap IDs instead of arrows...*

...but stop when you get tired or your pencil starts melting.

```
1  def icarus(x: int) -> int:  
2      """Unbounded aspirations!"""  
3      print(f"Height: {x}")  
4      return icarus(x=x + 1)  
5  
6  
7  print(icarus(x=0))
```

Questions to discuss with a neighbor:
What seems *wrong* with this function?
How might you *prevent* it?

Base Cases vs. Recursive Cases

Writing recursive functions that are non-infinite...

Trace the Following Program in a Diagram

```
1  def safe_icarus(x: int) -> int:
2      """Bounded aspirations!"""
3      if x >= 2:
4          return 1
5      else:
6          return 1 + safe_icarus(x=x + 1)
7
8
9  print(safe_icarus(x=0))
```

Parameters and Arguments

Keyword Arguments

Positional Arguments

Default Parameters

Trace the Following Program with Heap IDs

```
1  ✓ def gen(stop: int, acc: tuple[int, ...] = (), i: int = 0) -> tuple[int, ...]:  
2      """Generate a tuple from i to stop."""  
3  ✓      if i >= stop:  
4          return acc  
5  ✓      else:  
6          return gen(stop, acc + (i,), i + 1)  
7  
8  
9  print(gen(2))
```


Code Along: Magic 8 Ball