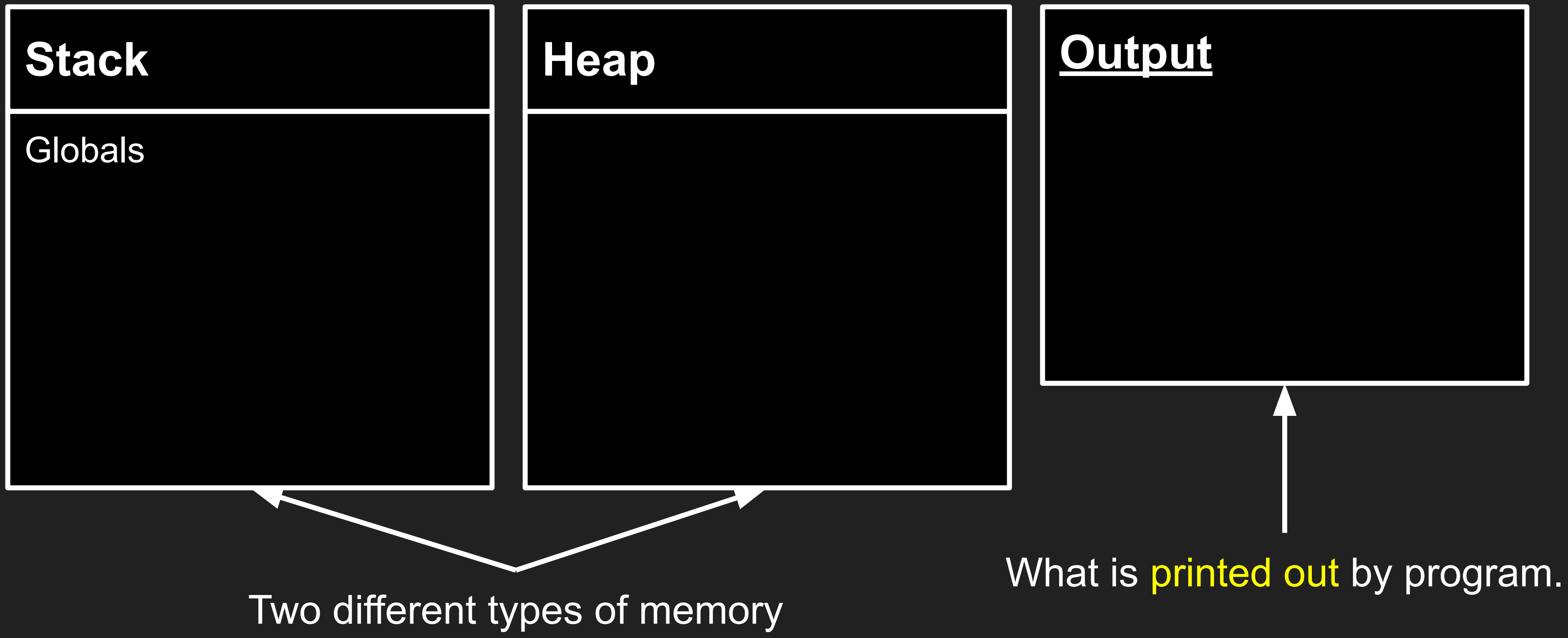
COMP 110

CL05: Memory Diagrams

Motivation

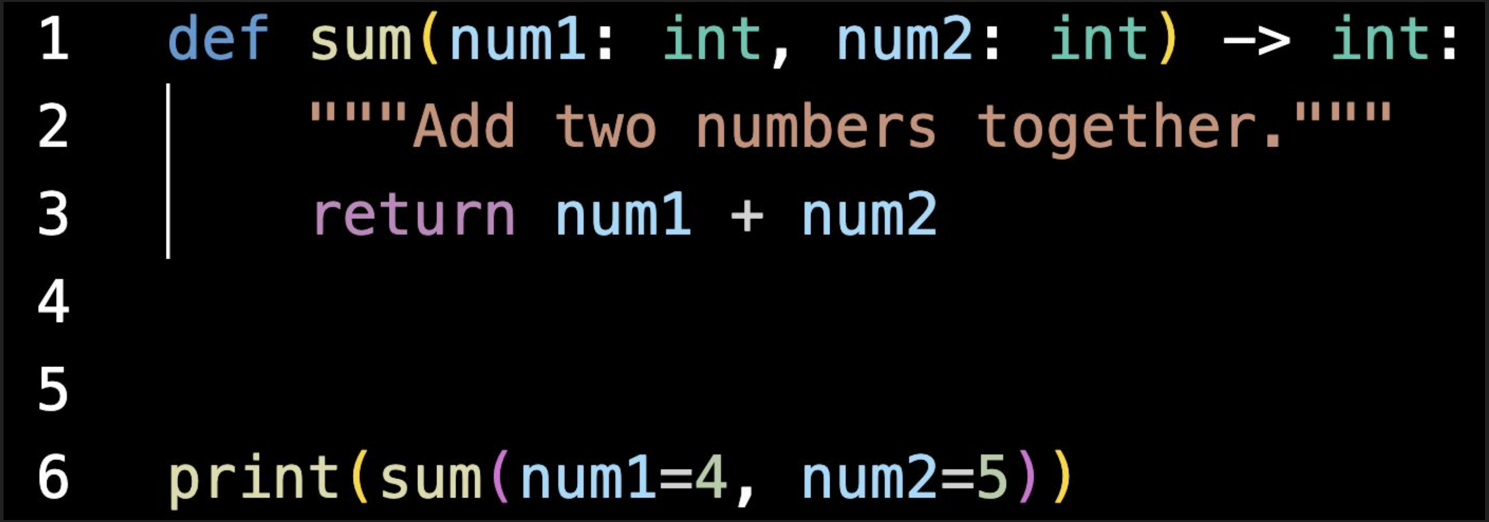
- Memory diagrams allow us to trace code in memory
- Helps us to understand what our code is doing and why

Memory Diagram Components


Stack
Globals
Heap
Output
Two different types of memory
What is printed out by program.

Stack vs. Heap

- Stack: variables, primitive types
- Heap: definitions, certain mutable types (more on this later)



```python
def sum(num1: int, num2: int) -> int:
    """Add two numbers together."""
    return num1 + num2


print(sum(num1=4, num2=5))
```

Stack

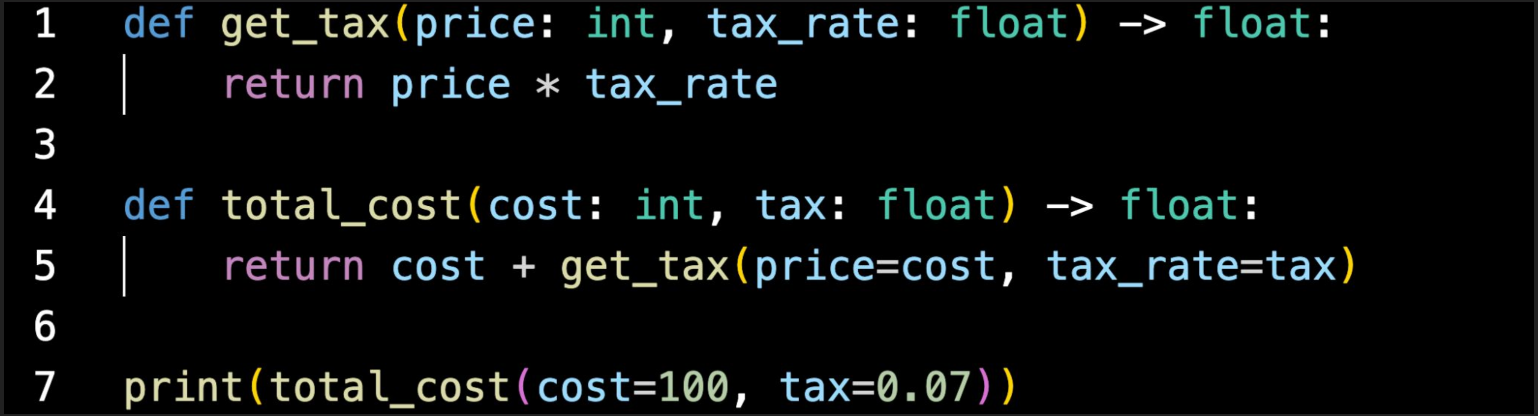
Globals

Heap

Output

Function Call Steps

- Prepare for call:
  - Has function been defined?
  - Are arguments fully evaluated?
  - Do parameters and arguments agree?
- Establish frame for function call:
  - Frame on stack labeled with function name
  - Return address
  - Copy over arguments



```python
def get_tax(price: int, tax_rate: float) -> float:
    return price * tax_rate

def total_cost(cost: int, tax: float) -> float:
    return cost + get_tax(price=cost, tax_rate=tax)

print(total_cost(cost=100, tax=0.07))
```

Stack

Globals

Heap

Output