CL07 - Dictionary + Scope Practice

Quiz 02!

- Working on creating a bank of extra (optional) practice questions to accompany the typical ones.
- Also will be extra (optional) coding practice questions. There will be a separate Gradescope you can join to check your answers with an autograder!
- *I will make a more detailed announcement about this via Sakai in the next few days!*

On the Horizon

CQ04

EX06

RD01

(Deadlines extended for break!)

```python
def f(x: float) -> float:
    x += 1.0
    y: float = x + 2.0
    return x + y

def g() -> None:
    global y
    x: float = f(3.0)
    y = f(x + 4.0)

x: float = 0.0
y: float = 0.0
g()
print(f"{x}, {y}")
```

```python
square_to_root: dict[int, int] = {}

i: int = 1
while i < 5:
    square_to_root[i ** 2] = i
    i += 1

print(square_to_root)
```

```python
square_to_root: dict[int, int] = {}

i: int = 1
while i < 5:
    square_to_root[i ** 2] = i
    i += 1

print(square_to_root)
```

```python
def main() -> None:
    names0: dict[str,str] = {"Pres.": "Lily", "VP.": "Ruby"}
    names1: dict[str,str] = {"Treas.": "Carlos", "Sec.": "Lin"}
    officers: dict[str,str] = merge(names0, names1)
    print(officers)

def merge(a: dict[str,str], b: dict[str,str]) -> dict[str,str]:
    result: dict[str,str] = {}
    for key in a:
        result[key] = a[key]
    for key in b:
        result[key] = b[key]
    return result

if __name__ == "__main__":
    main()
```

```python
def main() -> None:
    names0: dict[str,str] = {"Pres.": "Lily", "VP.": "Ruby"}
    names1: dict[str,str] = {"Treas.": "Carlos", "Sec.": "Lin"}
    officers: dict[str,str] = merge(names0, names1)
    print(officers)

def merge(a: dict[str,str], b: dict[str,str]) -> dict[str,str]:
    result: dict[str,str] = {}
    for key in a:
        result[key] = a[key]
    for key in b:
        result[key] = b[key]
    return result

if __name__ == "__main__":
    main()
```

Challenge Question (CQ04)

- Function name: zip
- Takes list[str] and list[int] as arguments
- Returns dict[str,int].
- The function should produce a dictionary where the keys are the items of the first list and the values are the corresponding items at the same index of the second list.
- If the input lists are different lengths or if they are empty, the function should return an empty dictionary.
- (More info on website!)