CS 105: Introduction to Computer Science

Prof. Thao Nguyen

Spring 2025

Materials adapted from Dave Wonnacott

Notecards

- Practice, group work: there will be a lot of this!
- Office hours: TA hours posted
- Visual demos, videos: I will try

Recap

- We distinguish *problem instances* from *general problems*, e.g.
    - problem instance: "multiply 17 by 38"
    - general problem: "multiply any two numbers"
- We answer a general problem with an *algorithm* (steps to solve any instance)
- We express ("encode") our algorithms with a *programming language*
    - e.g., a program can be expressed as one or more *Python functions*

Python Recap

What we've seen so far, of Python

- Calculations (+, -, >=, etc.)
- Variable definitions (e.g., x=12)
- Function definitions (with parameters, return)
- Conditional statements ("if", with test, true-part, false-part)
- Printing

Practice example

```
x = 2

y = 5
```

Swap values of x and y using only variables

Printing vs. returning

What are the results of each of the programs below?

```
def occupants(n_fac: int, n_stu: int) -> int:
    return n_fac + n_stu

print("during class, room has", occupants(1, 18), "people")
print("if half leave during break, about", occupants(1, 18) // 2)
```

```
def occupants(n_fac: int, n_stu:int) -> None:
    print("room contains", n_fac + n_stu, "people")

occupants(1, 18)
print("if half leave during break, about", occupants(1, 18) // 2)
```

Printing vs. returning (cont.)

What are the results of each of the programs below?

```
def occupants_and_ratio(n_fac: int, n_stu: int) -> float:
    return n_fac + n_stu
    return n_stu/n_fac

print("class size and stu/fac ratio are", occupants_and_ratio(1, 18))
```

→ “return” statement ends the function

```
def occupants_and_ratio(n_fac: int, n_stu: int) -> None:
    print("class size is", n_fac + n_stu)
    print("stu/fac ratio is", n_stu/n_fac)

occupants_and_ratio(1, 18)
```

Understanding Code Execution #1: the Debugger

PyCharm and PyCharm debugger demo

Some engineering terminology: "black-box" and "clear-box" analysis

Understanding Code Correctness #1: Test Suites

DocTest test suite demo in PyCharm

Discussion of the finicky nature of programming, and especially DocTest

Practice writing tests for division function

Scope

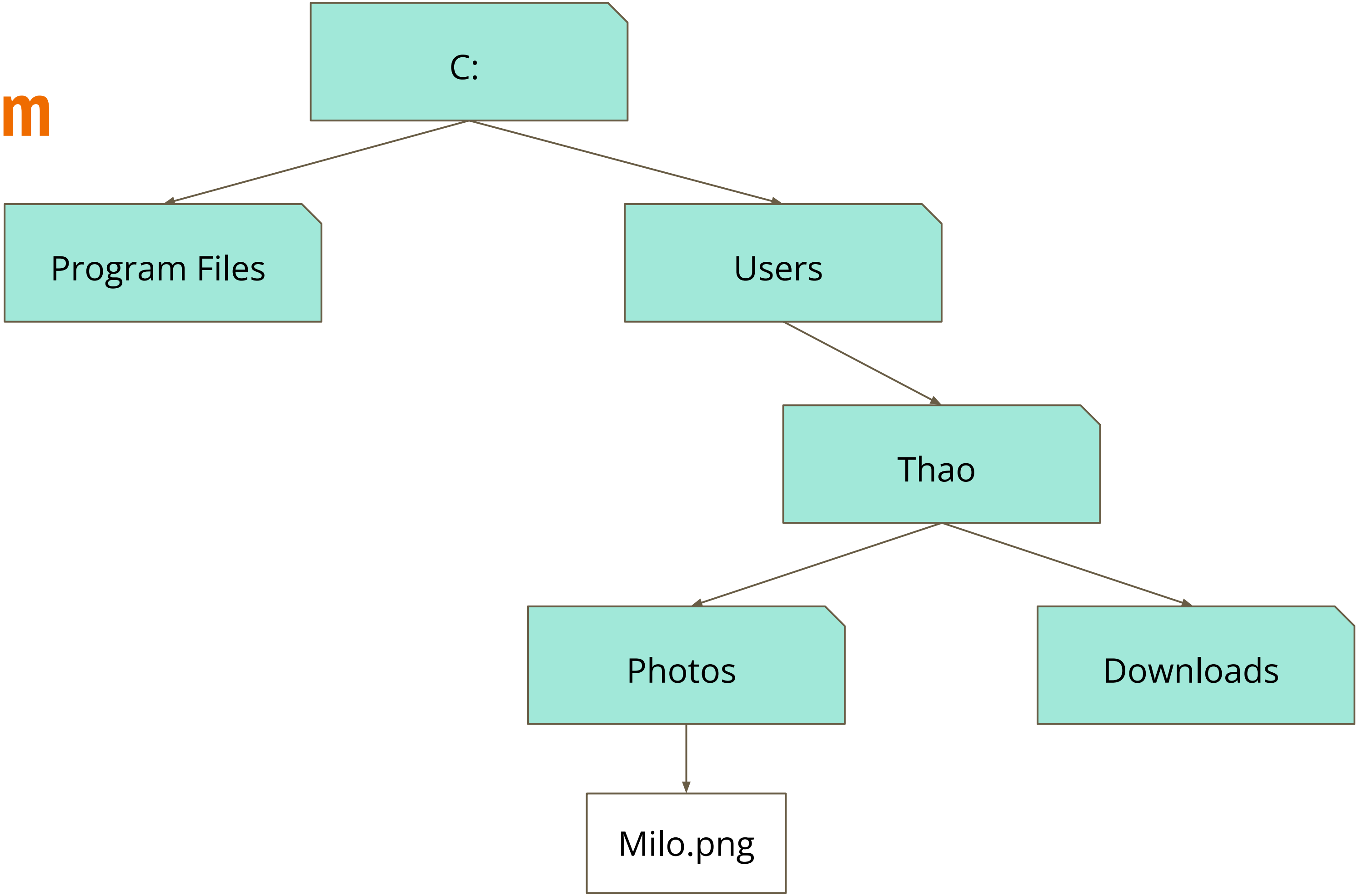
Local scope:

A variable created inside a function belongs to the *local scope* of the function, and is only available inside that function.

Global scope:

A variable created in the main body of the Python code is a global variable and belongs to the *global scope*. Global variables are available from any scope.

File System



Command line cheat sheet

Change directory to <directory>

```
$cd <directory>
```

Navigate to parent directory

```
$cd ..
```

Display path to current working directory

```
$pwd
```

List directory contents

```
$ls
```

Create new directory named <directory>

```
$mkdir <directory>
```

Computational Geometry Lab Problem Introduction

OverlapTestGraphics.py demo