3. print sep = end =
    a. With newline
    b. See program 01-10-lookAtPrint.py

4. Best Practice – Run often – "Islands of good code"

5. Errors
    c. Syntax error
        i. Examples
            1. "Print" vs "print"
            2. Two prints on same line
            3. No close quote in input
            4. Misspelled variable names: area vs. Area
        ii. Easiest to fix. Python tells you where to look

    d. Runtime error
        i. Examples
            1. height and width as strings → area = height * width
            2. divide by zero
        ii. Harder to find, still not too bad. Where did your program crash?

    e. Logic error
        i. Examples
            1. height and width are strings
                a. perimeter = height + height + width + width
            2. Hardest to find. Program runs, no crash, wrong results given

Make sure turtle PDF handout is posted ahead of time!

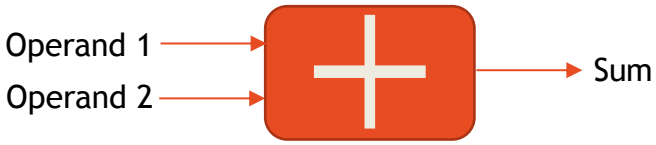
Section 1.5.3, Abstraction (Miller 3rd ed)

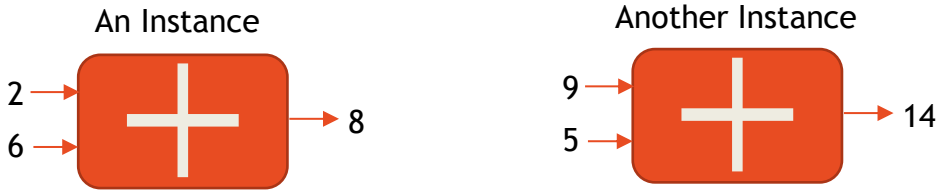
1. Abstraction
    a. "A concept unrelated to a specific instance"

2. In Real Live – A car
    a. Complexity abstracted to
        i. Steering Wheel
        ii. Gas pedal
        iii. Brake pedal
    b. This abstraction works
        i. For any car
        ii. Not just a specific car

3. The "+" key on a calculator
    a. Takes the same steps for all pairs of numbers

4. Functions are like a "Black Box"



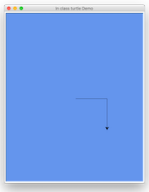


5. You've seen "Functional Abstraction" in Python
    a. The print() function - All of the work of
        i. Converting everything to a string
        ii. Connecting your program to the OS Standard Output Device
        iii. Sending the right Unicode sequences to the OS
        iv. Etc.

    b. All that is abstracted to the single function call: print()

The turtle Module

1. First Look at Python turtle
    a. 01-11-turtleDemo-1-Basic.py



    b. Colors: https://www.tcl.tk/man/tcl8.4/TkCmd/colors.htm