Week 4: HOFs & Environment Diagrams
Higher Order Functions: Returning a New Function
Learning Objectives

• Learn how to use and create higher order functions:
• Functions can be used as data
• Functions can accept a function as an argument
• Functions can return a new function
Review: What is a Higher Order Function?

• A function that takes in another function as an argument

OR

• A function that returns a function as a result.
• A function that returns (makes) a function

```python
def leq_maker(c):
    def leq(val):
        return val <= c
    return leq
```

```python
>>> leq_maker(3)
<function leq_maker.<locals>.leq at 0x1019d8c80>
```

```python
>>> leq_maker(3)(4)
False
```

```python
>>> [x for x in range(7) if leq_maker(3)(x)]
[0, 1, 2, 3]
```
Demo
Environment Diagrams
Environment Diagrams

• Organizational tools that help you understand code

• Terminology:
  – **Frame**: keeps track of variable-to-value bindings, each function call has a frame
  – **Global Frame**: global for short, the starting frame of all python programs, doesn’t correspond to a specific function
  – **Parent Frame**: The frame of where a function is defined (default parent frame is global)
  – **Frame number**: What we use to keep track of frames, f1, f2, f3, etc
  – **Variable** vs **Value**: x = 1. x is the variable, 1 is the value
Environment Diagrams Reminders

1. Always draw the global frame first
2. When evaluating assignments (lines with single equal), always evaluate right side first
3. When you CALL a function MAKE A NEW FRAME!
4. When assigning a primitive expression (number, boolean, string) write the value in the box
5. When assigning anything else (lists, functions, etc.), draw an arrow to the value
6. When calling a function, name the frame with the intrinsic name – the name of the function that variable points to
7. The parent frame of a function is the frame in which it was defined in (default parent frame is global)
8. If the value for a variable doesn’t exist in the current frame, search in the parent frame
Demo

Example 1:
• Primitives and Functions: Environment Diagram Python Tutor:

Example 2:
• make_adder Higher Order Function: Environment Diagram Python Tutor Link

Example 3:
• Compose Python Tutor Link
Example 1

```python
a = "chipotle"
b = 5 > 3
c = 8

def foo(c):
    return c - 5

def bar():
    if b:
        a = "taco bell"

result1 = foo(10)
result2 = bar()
```
Example 2

def make_adder(n):
    def adder(k):
        return k + n
    return adder

n = 10
add_2 = make_adder(2)
x = add_2(5)
add_2 = make_adder(2)
add_3 = make_adder(3)

x = add_2(2)
def compose(f, g):
    def h(x):
        return f(g(x))
    return h

add_5 = compose(add_2, add_3)
z = add_5(x)
Environment Diagram Tips / Links

• NEVER draw an arrow from one variable to another.

• Useful Resources:
  – http://albertwu.org/cs61a/notes/environments.html