Discussion 2:

Environment Diagrams + Hofs + Exam Prep (Oh my!)

Caroline Lemieux (clemieux@berkeley.edu)

February 7th, 2019

Administrivia

Administrativa

Homeworks

HW 2 due tomorrow at 11:59 PM

Projects

Hog due tonight at 11:59 PM

CSM

Sign up by Friday

Midterm 1

Monday 2/11, 7-8PM

HKN Review Session Saturday 2/9 12-3 PM in HP Auditorium

CSM Review Session Sunday 2/10 2-4 PM in GPB100

No lab next week (2/11-2/13)!

Agenda

I. Administrativa
II. HOF review & more environment diagram practice
III. Exam Tips & A Practice Problem

Environment diagram + HOF Review

Recap Quiz! (not for grades)

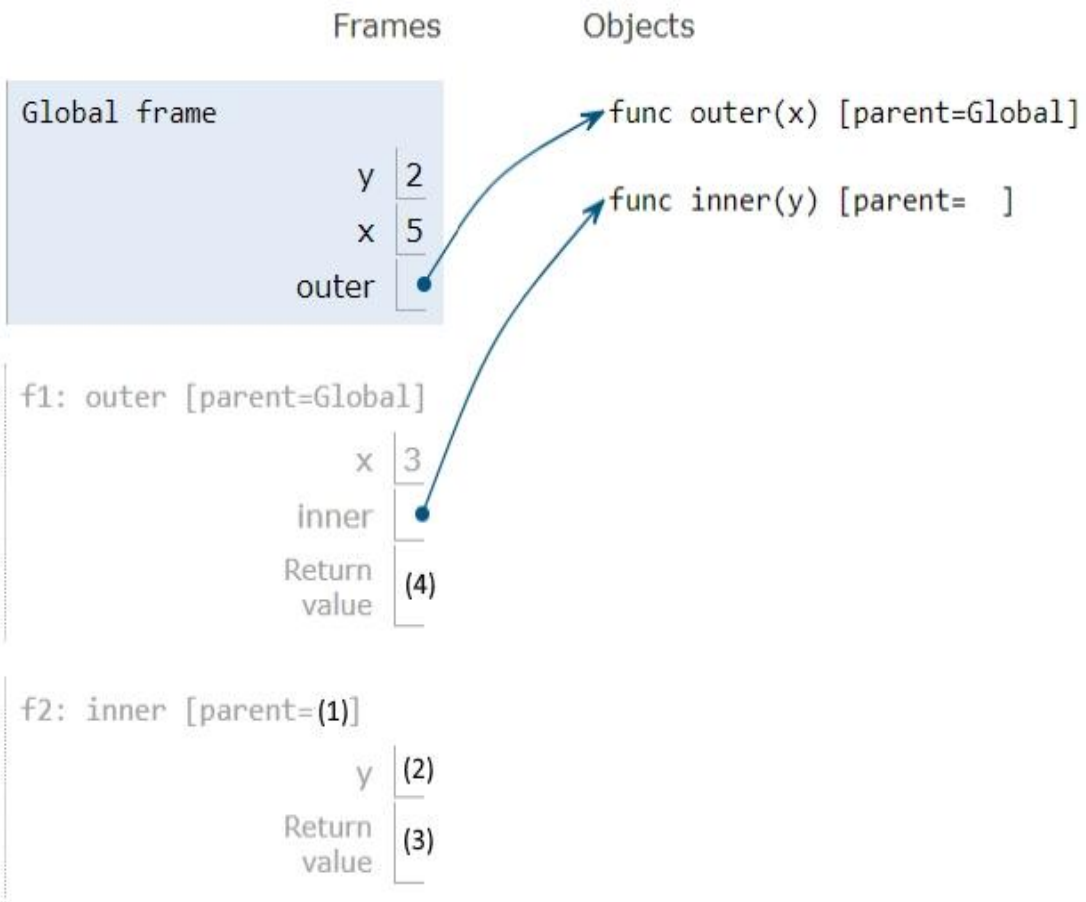
Participate online:

https://www.socrative.com/

Student login → room: CARO61A

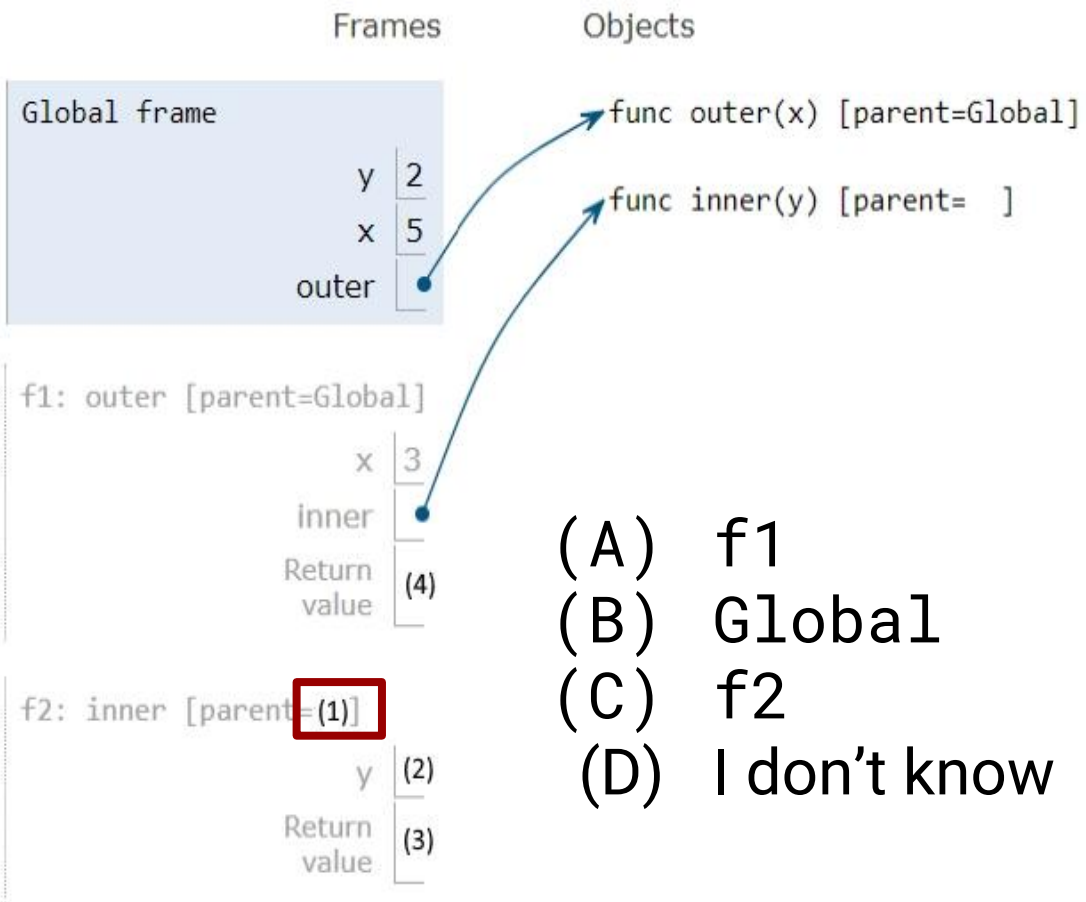
Python 3.6

```
1  y = 2
2  x = 5
3
4  def outer(x):
5      def inner(y):
6          return x + y
7      return inner(x)
8
9  outer(3)
```



Python 3.6

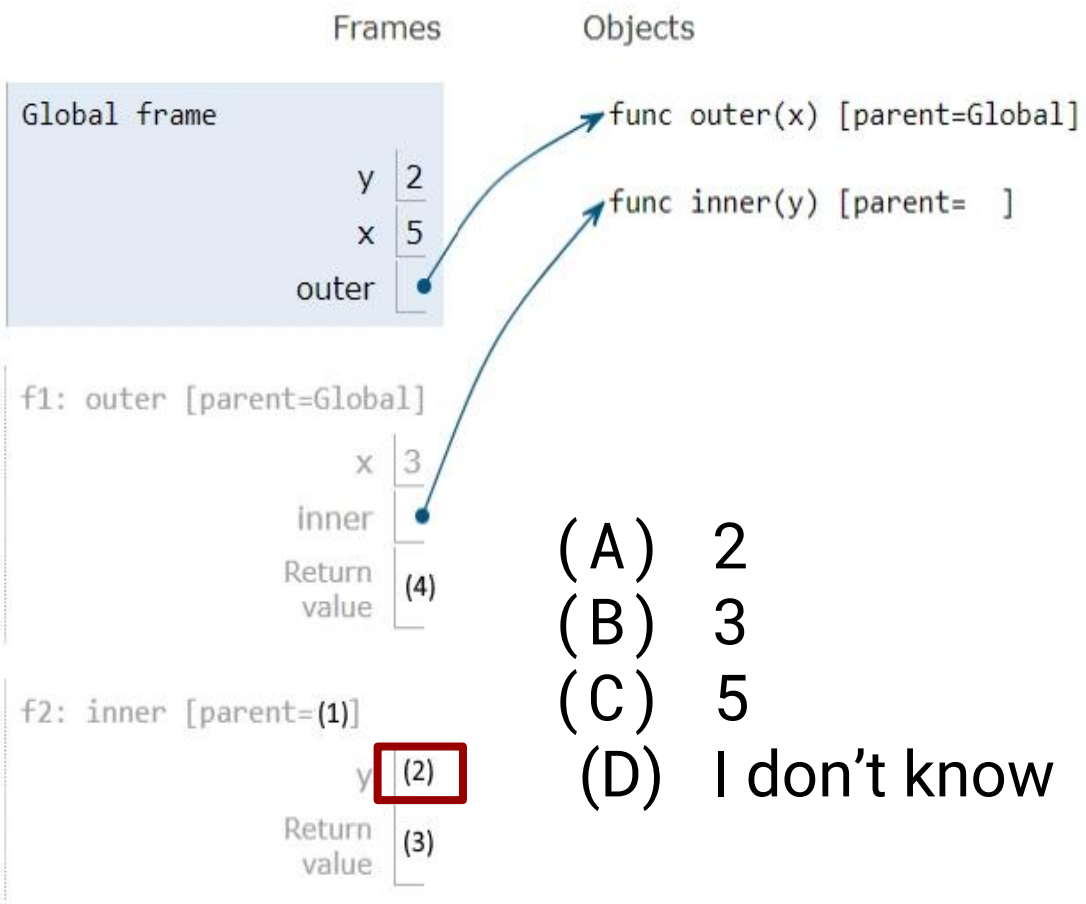
```
1  y = 2
2  x = 5
3
4  def outer(x):
5      def inner(y):
6          return x + y
7      return inner(x)
8
9  outer(3)
```



(A) f1
(B) Global
(C) f2
(D) I don't know

Python 3.6

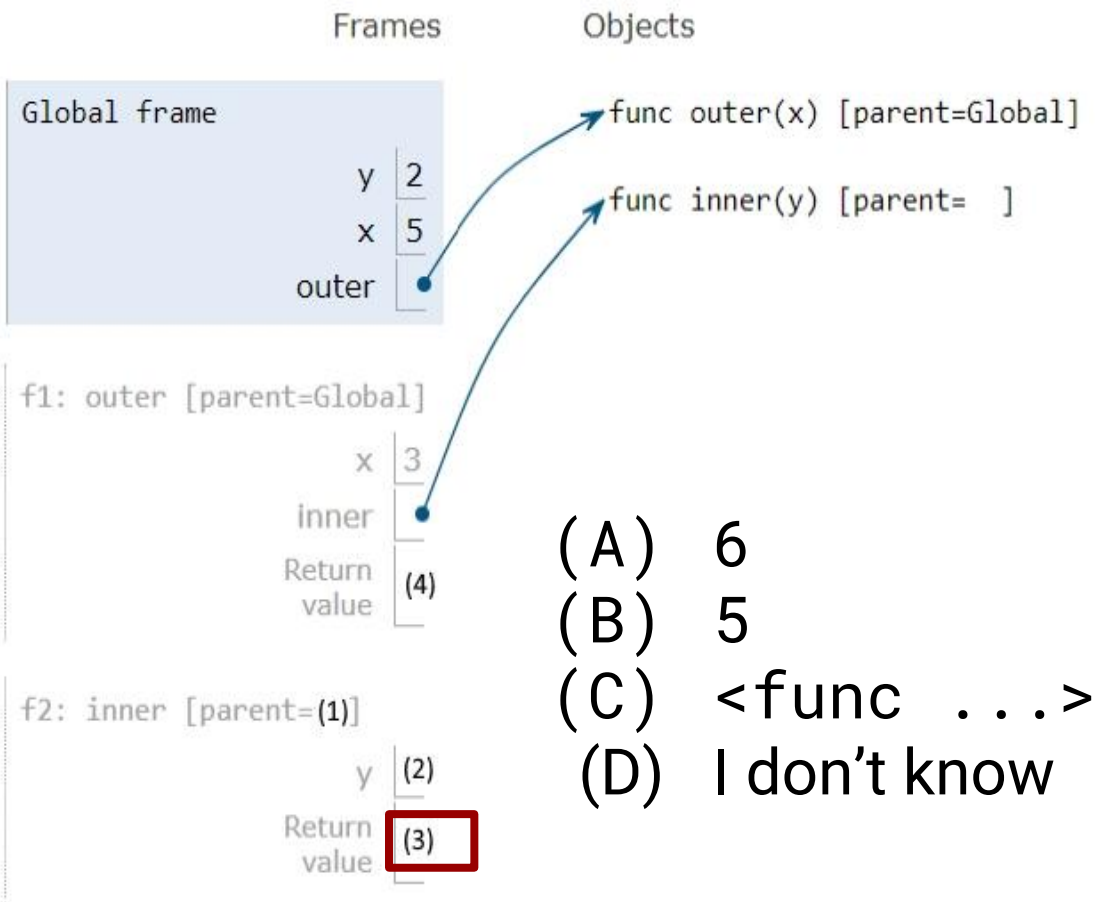
```
1  y = 2
2  x = 5
3
4  def outer(x):
5      def inner(y):
6          return x + y
7      return inner(x)
8
9  outer(3)
```



(A) 2
(B) 3
(C) 5
(D) I don't know

Python 3.6

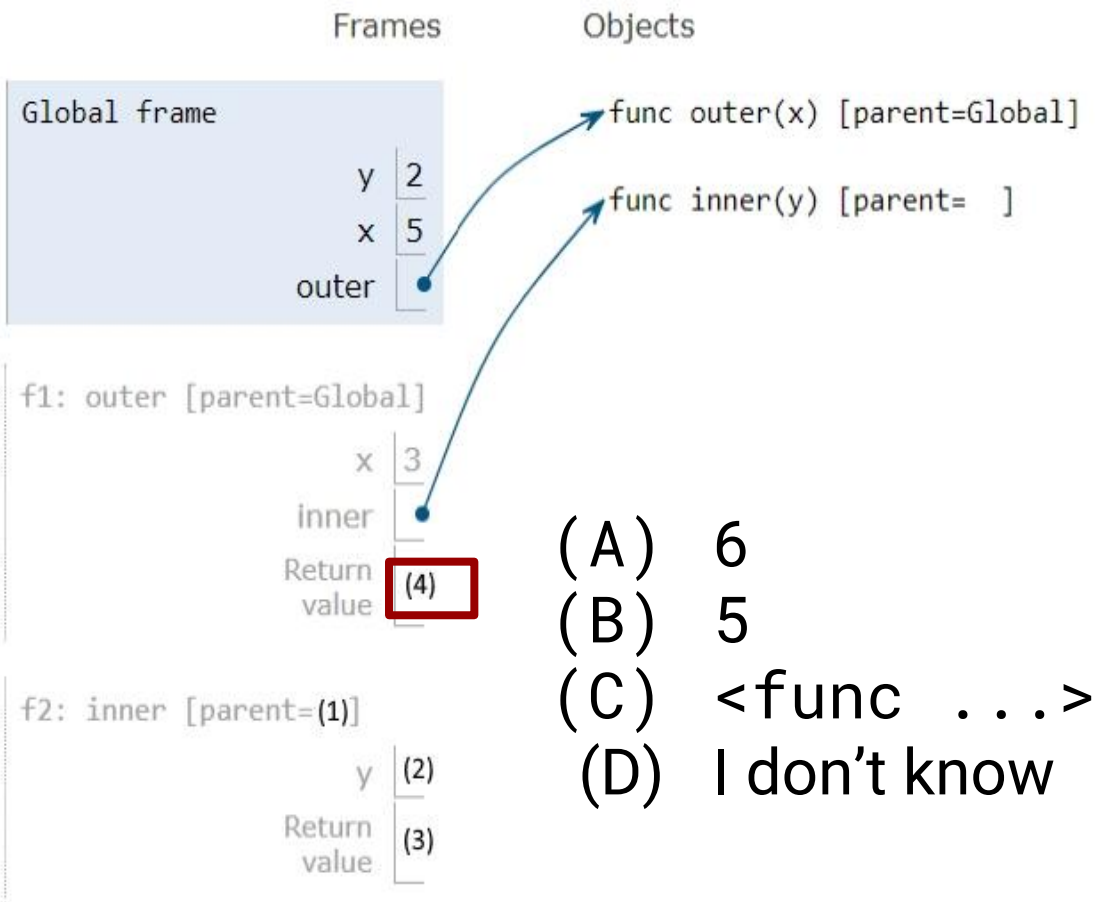
```
1  y = 2
2  x = 5
3
4  def outer(x):
5      def inner(y):
6          return x + y
7      return inner(x)
8
9  outer(3)
```



(A) 6
(B) 5
(C) <func ...>
(D) I don't know

Python 3.6

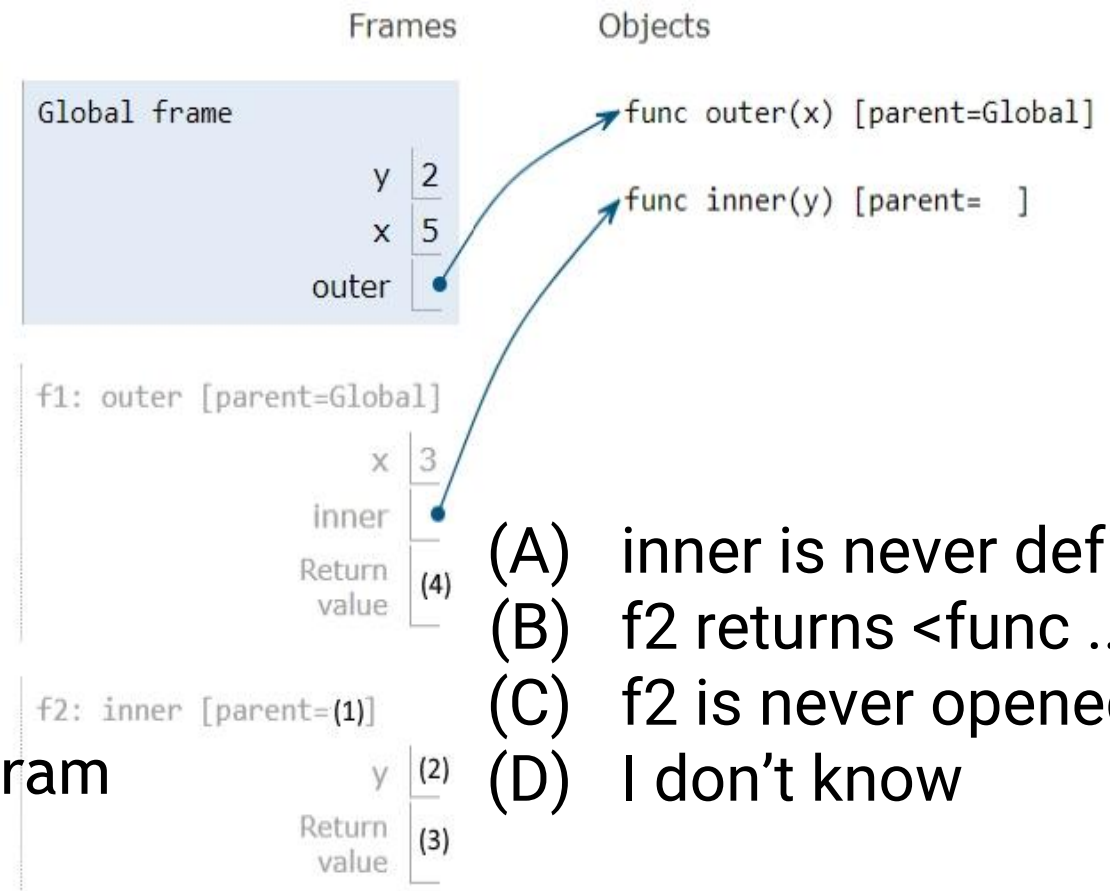
```
1  y = 2
2  x = 5
3
4  def outer(x):
5      def inner(y):
6          return x + y
7      return inner(x)
8
9  outer(3)
```



(A) 6
(B) 5
(C) <func ...>
(D) I don't know

Python 3.6

```
1  y = 2
2  x = 5
3
4  def outer(x):
5      def inner(y):
6          return x + y
7      return inner
8
9  outer(3)
```

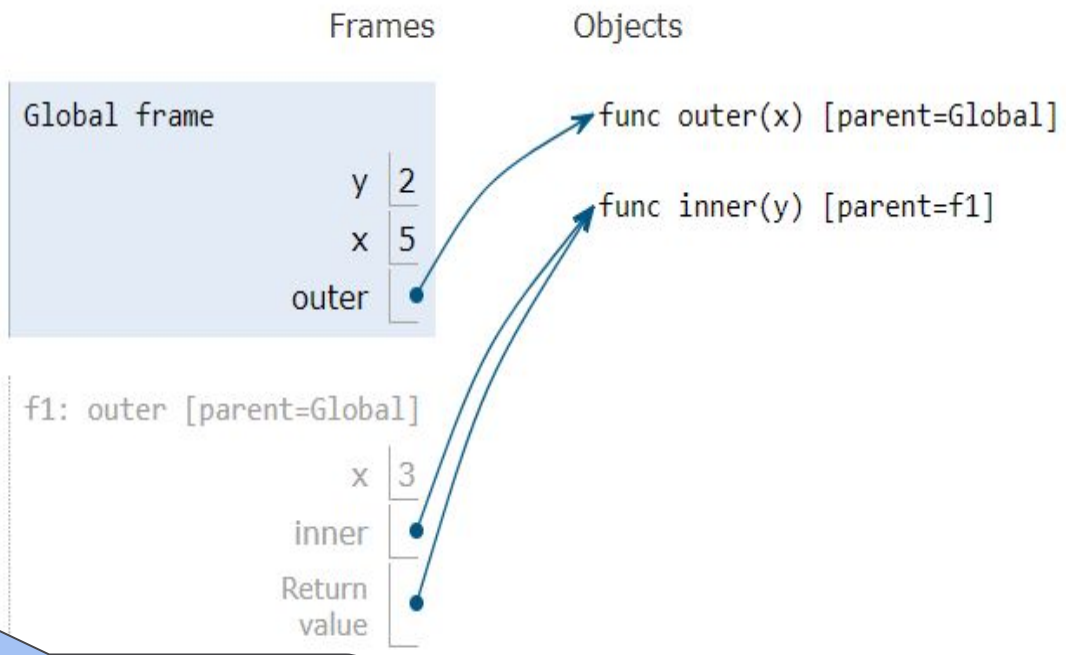


What does the environment diagram look like now?

(A) inner is never defined
(B) f2 returns <func ...>
(C) f2 is never opened
(D) I don't know

Python 3.6

```
1  y = 2
2  x = 5
3
4  def outer(x):
5      def inner(y):
6          return x + y
7      return inner
8
9  outer(3)
```



Don’t call inner, so we just return a ***function value***

Function Values

- Executing a def statement or evaluating a lambda creates a ***function value***
  - Function values = a value that can be called later
  - The result of calling a function value is what that function returns
  - But a function value doesn’t access its body until called!

HOFs

- Higher order functions (HOFs) either take in a function value or return a function value
  - We just did an example of this!
- Pay attention to return types and input types
  - Does a function return another function?
  - If so, how many parameters for that function? What does that function return?

Try Problem 1.5!

Try Problem 1.6!

Attendance

links.cs61a.org/caro-disc

Review: Lambdas

- **Expressions** that evaluate to **function values**
  - Don’t access their return values until you call them
- Bodies only have a single expression that you return
- Don’t have an intrinsic name

Review: Lambdas

```
lambda x, y: x + y
```

Review: Lambdas

```
lambda x, y: x + y
```

Takes in parameters x + y

Review: Lambdas

```
lambda x, y: x + y
```

When called,
returns x plus y

Try Problem 1.2!

Lambdas in Environment Diagrams

```
(lambda x, y: x + y)(1, 2)
```

Lambdas in Environment Diagrams

```
(lambda x, y: x + y)(1, 2)
```

Evaluate operator to get a function value

Lambdas in Environment Diagrams

```
(lambda x, y: x + y)(1, 2)
```

Evaluate each operand

Lambdas in Environment Diagrams

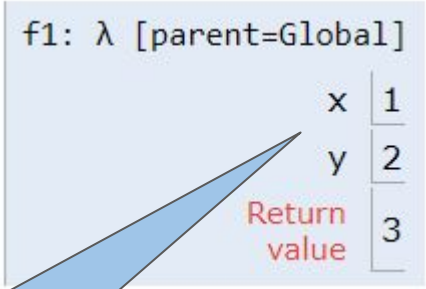
```
(lambda x, y: x + y)(1, 2)
```

```
f1: λ [parent=Global]
```

Create a new frame,
no intrinsic name.
Parent is where
lambda was evaluated

Lambdas in Environment Diagrams

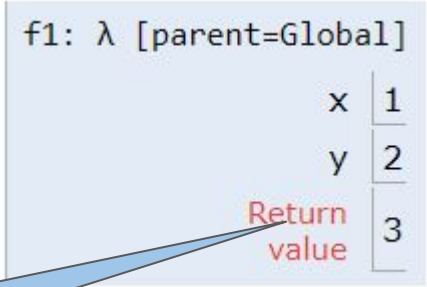
```
(lambda x, y: x + y)(1, 2)
```



Bind arguments to parameters, as before

Lambdas in Environment Diagrams

```
(lambda x, y: x + y)(1, 2)
```



Evaluate body of lambda and return that

Lookups

```
def inner(y):
    return x + y
```

f2: inner [parent=f1]

y	10
Return value	13

To find the value of a variable not in current frame, look in parent frame

Lambdas

Evaluates to a function value

```
lambda x, y: x + y
```

parameters

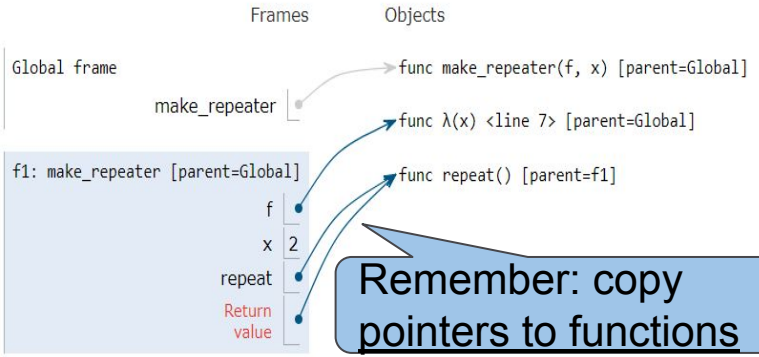
Expression returned (after evaluating)

HOFs

Take in a function value or return one

Python 3.6

```
1 def make_repeater(f, x):
2     def repeat():
3         f(x)
4         f(x)
5     return repeat
6
7 make_repeater(lambda x: print(x), 2)
```



Remember: copy pointers to functions

Try Problem 1.1!

Try Problem 1.3!

Exam Problem

Overview

1. WWPD
2. Environment Diagrams
3. Code Writing Question

Overview

1. WWPD
2. Environment Diagrams
3. Code Writing Question

These problems might use ***everything*** we've seen so far!

Copy this skeleton:

```
def differs_by_one_digit(m, n):
    diffs = 0
    while m > 0:
        if ____________________________________:
            return False
        m , t = m // 10 , m % 10
        n , v = n // 10 , n % 10
        if ____________________________________:
            diffs = ___________________________
    return _________________
```

Fill in the blanks of the implementation of differs_by_one_digit below, a function that takes two positive integers m and n and returns whether m and n differ in exactly one digit. If m and n have different numbers of digits, then differs_by_one_digit(m, n) always returns False. (assume m, n are positive integers)

```
>>> differs_by_one_digit(3467, 3427) # 3rd digit differs
True
>>> differs_by_one_digit(2013, 2011) # Last digit differs
True
>>> differs_by_one_digit(1013, 2013) # First digit differs
True
>>> differs_by_one_digit(5, 2)       # Only digit differs
True
>>> differs_by_one_digit(2013, 2013) # No digit differs
False
>>> differs_by_one_digit(1013, 2011) # Both first and last differ
False
>>> differs_by_one_digit(3102, 2013) # All digits differ
False
>>> differs_by_one_digit(1, 21)      # Different number of digits
False
>>> differs_by_one_digit(1, 12)      # Different number of digits
False
>>> differs_by_one_digit(21, 1)      # Different number of digits
False
>>> differs_by_one_digit(12, 1)      # Different number of digits
False
"""
```