LAMBDAS

COMPUTER SCIENCE MENTORS CS 88

February 22 to 26

1 Lambdas

A lambda expression evaluates to a function, called a lambda function. For example, `lambda x, y:  x + y` is a lambda expression, and can be read as "a function that takes in two parameters `x` and `y` returns `x + y`."

A lambda expression by itself evaluates to a function but does not bind it to a name. Also note that the return expression of this function is not evaluated until the lambda is called. This is similar to how defining a new function using a def statement does not execute the function's body until it is later called.

```
>>> what = lambda x : x + 5
>>> what
<function <lambda> at 0xf3f490>
```

Unlike `def` statements, lambda expressions can be used as an operator or an operand to a call expression. This is because they are simply one-line expressions that evaluate to functions.

```
>>> (lambda y: y + 5)(4)
9
>>> (lambda f, x: f(x))(lambda y: y + 1, 10)
11
```

1. What do lambda expressions do? Can we write all functions as lambda expressions? (Hint: think about the limitations of lambdas) In what cases are lambda expressions useful?

2. Determine if each of the following will error:

```
>>> 1/0

>>> boom = lambda: 1/0

>>> boom()
```

3. Express the following lambda expression using a **def** statement, and the **def** statement using a lambda expression.

```
pow = lambda x, y: x**y
```

```
def foo(x):
    def f(y):
        def g(z):
            return x + y * z
        return g
    return f
```

4. For each of the following lines of code, determine what would be printed as the output.

```
>>> plus_one = lambda i: print(i+1)
>>> plus_one


>>> plus_one(6)


>>> multiply = lambda x, y: x*y
>>> harder_lambda = lambda func: print(func(4, 5))
>>> harder_lambda(multiply)
```

5. What would Python print?

```
>>> a = lambda: 5
>>> a()

>>> a(5)

>>> b = lambda: lambda x: 3
>>> b()(15)

>>> c = lambda x, y: x + y
>>> c(4, 5)

>>> d = lambda x: lambda y: x * y
>>> d(3)

>>> d(3)(3)

>>> e = d(2)
>>> e(5)

>>> f = lambda: print(1)

>>> g = f()
```

6. **Challenge Problem**: Draw Environment Diagrams for the following lines of code. Note: When working with lambdas in environment diagram problems, it is really helpful to write down which line the lambda was defined on.

```
square = lambda x: x * x
higher = lambda f: lambda y: f(f(y))
b = higher(square)(5)
a = (lambda f, a: f(a))(lambda b: b * b, 2)
```

7. *The following question is extremely difficult. Something like this would not appear on the exam. Nonetheless, it's a fun problem to try.*

   Draw the environment diagram that results from executing the code below.

   Note that using the + operator with two strings results in the second string being appended to the first. For example `"C" + "S"` concatenates the two strings into one string `"CS"`

```
y = "y"
h = y
def y(y):
    h = "h"
    if y == h:
        return y + "i"
    y = lambda y: y(h)
    return lambda h: y(h)
y = y(y)(y)
```