ENVIRONMENT DIAGRAMS & HIGHER-ORDER FUNCTIONS Solutions

COMPUTER SCIENCE MENTORS 61A

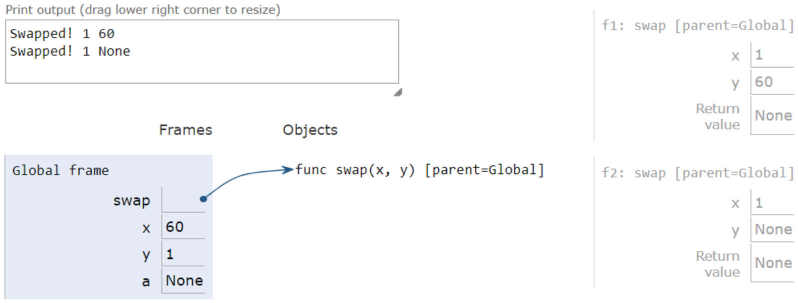
September 9 – September 13, 2024

1 Environment Diagrams

1. Give the environment diagram and console output that result from running the following code.

```python
def swap(x, y):
    x, y = y, x
    return print("Swapped!", x, y)

x, y = 60, 1
a = swap(x, y)
swap(a, y)
```



Print output (drag lower right corner to resize)

```
Swapped! 1 60
Swapped! 1 None
```

https://tinyurl.com/y68m6qdj

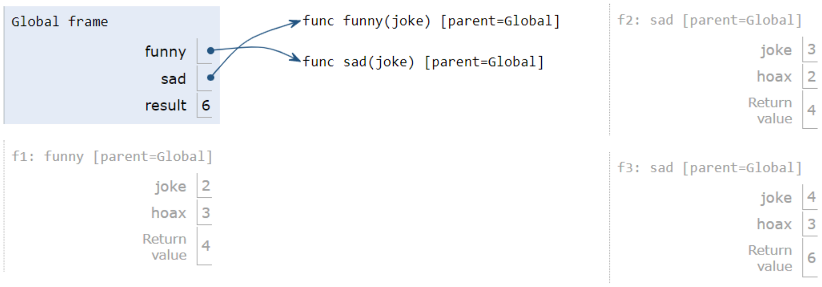
Created by Jim Fang, Bryan Adolfo Ventura Benitez. Worksheet feedback: links.jimfang.dev/feedback

2. Draw the environment diagram that results from running the following code.

```
def funny(joke):
    hoax = joke + 1
    return funny(hoax)

def sad(joke):
    hoax = joke - 1
    return hoax + hoax

funny, sad = sad, funny
result = funny(sad(2))
```



Global frame
funny
sad
result 6

func funny(joke) [parent=Global]
func sad(joke) [parent=Global]

f1: funny [parent=Global]
joke 2
hoax 3
Return value 4

f2: sad [parent=Global]
joke 3
hoax 2
Return value 4

f3: sad [parent=Global]
joke 4
hoax 3
Return value 6

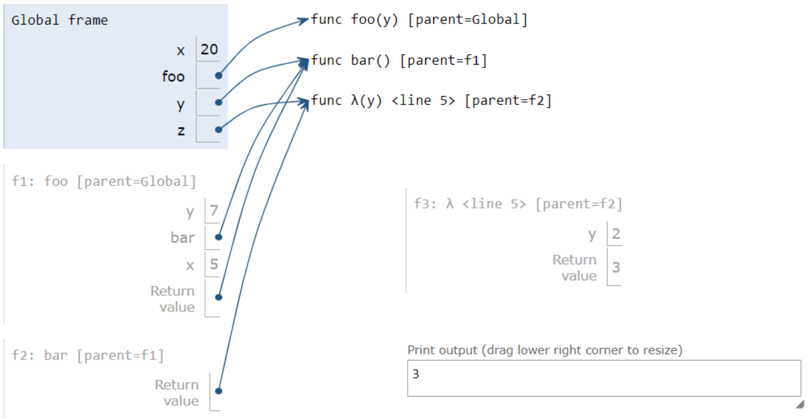
https://tinyurl.com/y5lc4fez

2 Higher-Order functions

1. Give the environment diagram and console output that result from running the following code.

```
x = 20
def foo(y):
    x = 5
    def bar():
        return lambda y: x - y
    return bar

y = foo(7)
z = y()
print(z(2))
```



Global frame
x 20
foo
y
z

func foo(y) [parent=Global]
func bar() [parent=f1]
func λ(y) <line 5> [parent=f2]

f1: foo [parent=Global]
y 7
bar
x 5
Return value

f2: bar [parent=f1]
Return value

f3: λ <line 5> [parent=f2]
y 2
Return value 3

Print output (drag lower right corner to resize)

```
3
```

https://tinyurl.com/yxfcvxxa

2. Fill in the blanks (*without using any numbers in the first blank*) such that the entire expression evaluates to 9.

```
(lambda x: lambda y: lambda: y(x))(3)(lambda z: z*z)()
```

3. Write a function, `whole_sum`, which takes in an integer, `n`. It returns another function which takes in an integer, and returns `True` if the digits of that integer sum to `n` and `False` otherwise.

```
def whole_sum(n):
    """
    >>> whole_sum(21)(777)
    True
    >>> whole_sum(142)(10010101010)
    False
    """
    def check(x):

        ___________________

        while _____________:

            last = __________________

            _________________________

            _________________________

        return __________________

    return _________________
```

```
def whole_sum(n):
    def check(x):
        total = 0
        while x > 0:
            last = x % 10
            x = x // 10
            total += last
        return total == n
    return check
```

4. Write a higher-order function that passes the following doctests.

   Challenge: Write the function body in one line.

```
def mystery(f, x):
    """
    >>> from operator import add, mul
    >>> a = mystery(add, 3)
    >>> a(4) # add(3, 4)
    7
    >>> a(12)
    15
    >>> b = mystery(mul, 5)
    >>> b(7) # mul(5, 7)
    35
    >>> b(1)
    5
    >>> c = mystery(lambda x, y: x * x + y, 4)
    >>> c(5)
    21
    >>> c(7)
    23
    """


    def helper(y):
        return f(x, y)
    return helper
```

   Challenge solution:

```
    return lambda y : f(x, y)
```

5. What would Python display?

   (a)
```
> (lambda x: x(x))(lambda y: 4)
```
```
4
```

   (b)
```
> (lambda x, y: y(x))(mul, lambda a: a(3, 5))
```
```
15
```