Higher-Order Functions: exam-level questions

> If you need help reviewing Higher-Order Functions, take a look at these resources:
>
> - Albert's and Robert's slides part 1 (https://docs.google.com/presentation/d/1RijXoFtQxe2zdl6dYzwn0KMTW8gXiP7MMQgKd-MHXBo/edit#slide=id.p)
> - Albert's and Robert's slides part 2 (https://docs.google.com/presentation/d/1K4a54Qp716fWcGaTLDAyAYv-CCBy4p6P66M97eTAFgI/edit#slide=id.p)
> - Albert's and Robert's slides part 3 (https://docs.google.com/presentation/d/11-75T8zaVP1V2rwADDDtyt77hX-LKIeUu1hPMmfGXME/edit#slide=id.ga271ab36d_1_364)

Each question has a "Toggle Solution" button -- click it to reveal that question's solution.

Environment Diagrams

Question 1

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

def g(x):
    return lambda : f(x) + f(y)

y = 2
result = f(g(f))
```

Toggle Solution

Question 2

```
def always_roll(n):
    return lambda s0, s1: n

def make_bad_strategy(p):
    def strategy(s0, s1):
        # next line is bad style!
        return always_roll(1 - p)(s0, s1)
    return strategy

num_rolls = make_bad_strategy(1)(50, 50)
```

Toggle Solution

Question 3

```
def test(fn):
    def new_fn(x):
        if x > 10:
            return new_fn(x % 10)
        else:
            return fn(x)
    return new_fn
x = 10
new = test(lambda score: score - x)
new(42)
```

Toggle Solution

Question 4

```
x = 4
def foo(foo):
    def bar(y):
        y += foo
        return lambda : y + x
    foo += 3
    return bar
x = 5
foo(5)(4)()
```

Toggle Solution

Question 5

```
def dream1(f):
    kick = lambda x: mind()
    def dream2(secret):
        mind = f(secret)
        kick(2)
    return dream2

inception = lambda secret: lambda: secret
real = dream1(inception)(42)
```

Toggle Solution

Question 6

```
def albert(albert):
    albert = albert()
    def albert():
        albert = lambda albert: albert
        return albert(albert)
    return albert

albert(lambda: albert)()
```

Toggle Solution