Expressions and Functions

What would Python print?

1. Order of evaluation:

```python
>>> def jurassic(park, world):
...     print(world)
...     return park - world

>>> def big(dino):
...     print(dino)
...     return 2 * dino
...     print(dino + 1)

>>> closed = jurassic(jurassic(5, 4), big(7))

>>> closed
```

1

```python```
2. print vs. return
   >>> x = print(42)

   >>> x

   >>> def foo(y):
   ...     return y * y
   >>> def bar(y):
   ...     print(y * y)
   >>> a = foo(4)
   >>> a == 16

   >>> b = bar(4)

   >>> b == 16

   >>> def garply(y):
   ...     print(y * y)
   ...     return 3
   >>> c = garply(4)

   >>> c
1. Implement `factorial(n)`, which takes a non-negative `n` and returns all the numbers from 1 to `n` multiplied together. For example, `factorial(5) = 1 * 2 * 3 * 4 * 5 = 120`.

*Note:* Your function should be able to compute `factorial(0)` to be 1, as defined in mathematics.

```python
def factorial(n):
    """Returns the product of numbers from 1 to n."

    >>> factorial(0)
    1
    >>> factorial(1)
    1
    >>> factorial(5)  # 1 * 2 * 3 * 4 * 5
    120
    """
```
1. Draw an environment diagram for the following code:

```python
x = 5

def illum(nati):
    y = nati + x
    return nati - x

def files(x):
    return illum(x) - x

x = files(6)
illum(4)
```

3 Higher order functions
2. Draw an environment diagram for the following code:

```python
def cons(piracy):
    def confirmed(x):
        return piracy(x + y)
    y = 4
    return confirmed

cons(lambda a: a + y)(5)
```