1 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))
```

Solution:

```
4
7
14
```

```python
>>> closed
```

Solution:

```
-13
```
2. print vs. return
   
   >>> x = print(42)

   Solution: 42

   >>> x

   Solution: Nothing shows up. This is because x is assigned to None (the return value of print)

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

   Solution: True

   >>> b = bar(4)

   Solution: 16

   >>> b == 16

   Solution: False

   Since bar does not have a return value, it implicitly returns None. Thus, b is assigned to None.

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

   Solution: 16

   >>> c

   Solution: 3
2 Control structures

1. Implement \texttt{factorial(n)}\texttt{}, which takes a non-negative \texttt{n} and returns all the numbers from 1 to \texttt{n} multiplied together. For example, \texttt{factorial(5) = 1 \times 2 \times 3 \times 4 \times 5 = 120}.

\textit{Note}: Your function should be able to compute \texttt{factorial(0)} to be 1, as defined in mathematics.

\begin{verbatim}
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
    """

\end{verbatim}

\textbf{Solution:}

\begin{verbatim}
i, total = 1, 1
    while i <= n:
        total = total * i
        i += 1
    return total
\end{verbatim}
1. Draw an environment diagram for the following code:

\[
\begin{align*}
\text{x} &= 5 \\
\text{def} \illumin (\text{nati}) : \\
&\quad y = \text{nati} + \text{x} \\
&\quad \text{return} \ nati - x \\
\text{def} \files(x) : \\
&\quad \text{return} \ \illumin(x) - x \\
\text{x} &= \files(6) \\
&\quad \illumin(4)
\end{align*}
\]

Solution:
2. Draw an environment diagram for the following code:

\[ y = 1 \]

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

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

Solution: