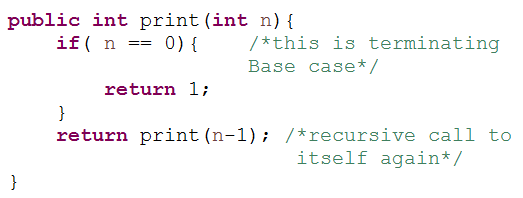
**Recursion and Memory (Visualization)**

* Each recursive call makes a new copy of that method (actually only the variables) in memory.
* Once a method ends (that is, returns some data), the copy of that returning method is removed from memory.
* The recursive solutions look simple but visualization and tracing takes times. For better understanding, let us consider the following example.



For this example, if we call the print function with n=4, visually our memory assignment may look like:

**print(3)**

**print(4)**

**print(1)**

**print(2)**

**print(0)**

**Returns 0**

**Returns 0**

**Returns 0**

**Returns 0**

**Returns 0 to main function**

Now, let us consider our factorial function. The visualization of factorial function with n=4 will look like:

**4!**

**4\*3!**

**Returns 4 \* 3!**

**3\*2!**

**Returns 3 \* 2!**

**2\*1!**

**Returns 2 \* 1!**

**1**

**Returns 1**

**Returns 24**