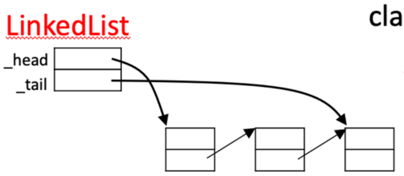
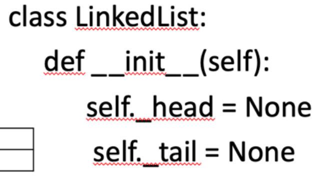
CSC 120 **ICA-16**

Work with your neighbor. (This will be graded for participation only.)

1. Suppose that we modify the linked list class to maintain a tail reference. The modified code for the `LinkedList` class and a sample linked list are shown below:





```
class LinkedList:
    def __init__(self):
        self._head = None
        self._tail = None
```

Write the `append(self, new)` method for the class.

2. Below are the method headers for a Stack class. Fill out the code to implement the methods of the Stack ADT. (*Note: given a list alist, the Python method alist.pop() removes the last element of the list.)*

```
class Stack:
    def __init__(self):
        self._items = 

    # adds item to the top of the Stack
    def push(self, item):



    # removes the top item from the Stack
    def pop(self):



    def is_empty(self):
```

Note: We may not get to these next 2 problems, in which case they will move to ICA-17.

3. Write a *function* `reverse(s)` that reverses the string `s` using a `Stack`. The function returns the reversed string.

4. Write a *function* `balanced(s)` that returns `True` if the string `s` is balanced with respect to the bracket characters `'['` and `']'` and `False` otherwise. Use a `Stack` in your implementation of this function.