

# Assignment #5: Ouroboros

**Due Date:**

## 1. Objective

The main purpose of this assignment is to have you all become familiar working with custom data structures that are programmer defined. In class, it has already been established that this is a huge benefit in the sense that it allows for a high degree of flexibility in how things can be approached and accomplished. This also serves as good practice in understanding how to handle and keep track of object references, which is key in the successful completion of this assignment.

## 2. Provided Files

There will be two Python files that will be needed in the successful completion of this assignment.

The first file is named **node.py**. This file contains the basic class definition for a Node (see Section 3- **Node Class Definition** for further information).

The second file is the usual tester program: **ouroborosTester.py**. More information on the tester program can be found in Section 6- **Testing**.

## 3. Node Class Definition

Recall that, to begin working with a linked list, a Node object must be first defined. Python supports a programming paradigm known as object-oriented programming. Below is the class definition for a Node in a singly linked list:

```
class Node(object):  
  
    def __init__(self, data, next = None):  
        self.data = data  
        self.next = next
```

Please note that you do **NOT** have to include the above class definition of a **Node**

within your program. The above definition can be found within the **node.py** Python file.

#### **IF YOU PLAN ON USING THE CLASS DEFINITION TO TEST YOUR FUNCTIONS**

**WITHOUT THE USE OF THE TESTER PROGRAM:** you will have to import specifically the **Node** class definition. This can be accomplished by including the following line of Python code at the top of your program:

```
from node import Node
```

From here, you can create **Node** objects simply by calling the name of the object (i.e. `var1 = Node(data)` as an example).

#### **4. Required Functions**

##### **insert\_and\_link(head, data)**

**Description:** Given the “head” of the Ouroboros data structure, and a piece of data (will always be an integer, and no other kind of data type), create and insert a new Node at the “tail” of the Ouroboros data structure while also linking the newly created Node (which is now the new “tail” of the Ouroboros data structure) to the “head” of the data structure.

**Output:** This function should not output anything to the screen.

**Returns:** This function should not return anything.

##### **remove\_and\_link(head)**

**Description:** Given the “head” of the Ouroboros data structure, remove the “tail” of the data structure. When the “tail” Node is removed, ensure to have the new “tail” Node link back to the “head” Node of the Ouroboros data structure.

**Output:** This function should not output anything to the screen.

**Returns:** This function should return the data stored within the node to be removed.

### **count\_nodes(head)**

**Description:** Given the “head” of the Ouroboros data structure, count every node comprising it up until the “tail” of the data structure is encountered.

**Output:** This function should not output anything to the screen.

**Returns:** This function should return the number of nodes that form part of the Ouroboros data structure.

## **5. Requirements**

Your program must be named “ouroboros.py”.

## **6. Testing**

The tester program will run a total of **9** test cases: **3** of the test cases will test the **insert\_and\_link()** function, **3** of the test cases will test the **remove\_and\_link()** function, and the remaining **3** test cases will test the **count\_nodes()** function.

The test cases are split into **3** groups, each one testing a different Ouroboros data structure. For all of the groups, the **insert\_and\_link()** function will be tested first. From there, either the **remove\_and\_link()** or the **count\_nodes()** function will be tested next, before testing the remaining function. In other words, if the **remove\_and\_link()** function is tested first, then the **count\_nodes()** function will be tested next, and vice-versa.

### **IF TESTING IS DONE ON REPL:**

I understand many of you will be using Repl on completing the assignment. As such, here is a suggested way of testing your program. Please note that this approach will **NOT** guarantee that it will work.

When a new Repl is created, the **main.py** file will be automatically created. Before moving forward, make sure that **THREE** files (excluding the **main.py** file) are to be present within the file hierarchy of the Repl:

the **node.py** file, the **ouroboros.py** file (THIS IS YOUR PROGRAM), and the **ouroborosTester.py** file. From this point, your Repl should now have a total of **FOUR** files in no particular order: the **main.py** file, the **node.py** file, the **ouroboros.py** file, and the **ouroborosTester.py**. Within the **main.py** file, include the following line of code:

```
import ouroborosTester
```

When you click **Run** on Repl, the tester program should then start running and providing feedback on your program.

This feedback will range from whether your program is not named correctly, or if its not found in the same folder as the other files, or if any of your functions are either not properly defined (in the sense of adhering to Python's syntax), or is misspelled, this will raise an error.

## 7. Help & Advice

If you feel overwhelmed by this assignment, don't be. A good starting point is to first create skeleton definitions of the required functions that returns dummy values. From there, work on the function that you feel is the easiest to implement. Once defined, run the **ouroborosTester.py** program and see if you were able to successfully implement it. From this point, rinse and repeat.

If you need any kind of clarification to anything on this assignment, such as if you're confused about the requirements for one of the functions, or if you encounter a bug error that you are unable to figure out on your own, please send an email to [chsprogvol@gmail.com](mailto:chsprogvol@gmail.com). You may expect a response within 24 to 48 hours