`UnorderedList` Performance

For the surrounding details regarding this programming assignment, refer to the lessons titled *Introduction to Algorithm Analysis*, *Basic Data Structures (Linked Lists)*, and *Basic Data Structures (Stacks)*.

Specification

Using the final `UnorderedList` class that we developed in one of the lessons, write a Python program that compares its performance with the Python list. To do this programming assignment properly, you will need to use the `UnorderedList` class that we developed in the *Stacks* lesson (which has the `pop()` function).

The first part of your program will test creating both lists **100** times. Each list should have **25,000** values, each value randomly chosen within the range 0 through 999. Similar to the examples in one of the lessons, use Python's `Timer` class (from the `timeit` library) to automate the process of testing the two data structures. To be clear, you should use the `Timer` class to time how long it takes to create each list of 25,000 values, 100 times.

The second part of your program will test deleting every item from each list. Since this process destroys the lists, use Python's `time()` function (from the `time` library) instead of the `Timer` class. Simply time how long it takes to delete every value from the lists – once.

Finally, display the results in a format similar to the sample output below.

Constraints and notes

Note the following constraints and notes (see the rubric below for more detail):

- Structure your output so that it is similar to the sample output shown below;
- Format floating point values to three places to the right of the decimal point;
- You must use good coding style (which includes things such as including an informative header at the top of your program, commenting your source code appropriately, using meaningful variable and (if applicable) constant identifiers, and so on);
- Use functions in a similar manner as the examples in the lessons when using the `Timer` class; and
- Ideally, also use functions to delete the values from each list.

Deliverable

Submit a Python 3 source file that can be executed through an IDE (such as Thonny) or via the command line (e.g., `python3 ListTests.py`).

Sample output

Here is sample output of one run:

```
Creating:
UnorderedList:  2.116s
  Python list:  1.151s
Deleting:
UnorderedList:  9.278s
  Python list:  0.002s
```

Hints

Here are some hints to help get you started:

- Of course, feel free to Google things and/or discuss high-level ideas with your classmates. As always, do not Google solutions and make sure to **cite sources** (including your classmates).
- The `timeit()` function returns the total time (in seconds) that it takes to create a list of 25,000 random values, 100 times. You are calculating the total time (not the average time).
- You can install the various library files that are used in this course (including those that are used in this programming assignment) by executing the following in a terminal: `pip install git+https://github.com/jgourd/CSC201UT`
- Then, import the library files for this programming assignment in your Python program as follows: `from CSC201UT import UnorderedList`

Rubric

Please note the following rubric for this programming assignment:

UnorderedList Performance		
#	**Item**	**Points**
1	Good coding style	5
2	Similar output	2.5
3	Number formats correct	2.5
4	Timer class properly used	8
5	Results are correct	7
TOTAL		**25**