**Course Project**

**DeVry University**

**College of Engineering and Information Sciences**

**Course Number: CEIS295**

# Module 2: LinkedList Real-World Speeds

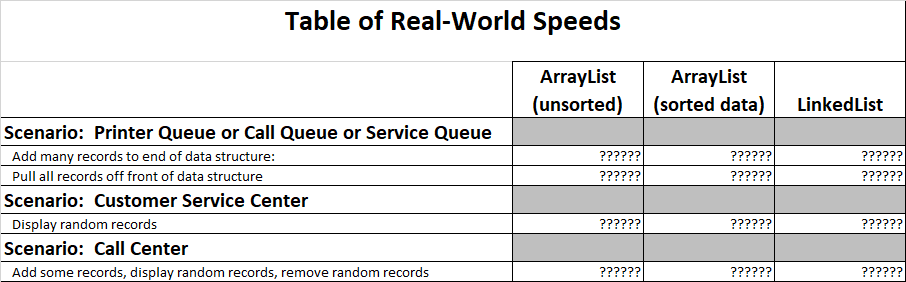
# Objectives

* To add Client data type to the LinkedList
* To develop Python code that tests the LinkedList’s real world speeds
* To update an Excel table that displays relevant real-world speeds

We are discussing the algorithm speeds using theoretical nomenclature, including Big-O notation. Theoretically, some algorithms are faster than other algorithms for certain tasks. Once you understand the theory, it is important to test the algorithm speeds using real-world data. How does the algorithm perform for different tasks? Let’s find out!

We are going to use the same time measurement technique to test the algorithms’ speeds this week that we used last week. In addition, we are going to test the algorithms using a same small, but real-world sized, dataset.

# Steps

1. Create a new folder in your CEIS295 folder called “Week 2 Project”. Copy the Excel table to this folder that we used last week so we can record the time that it takes to perform real-world processes based on common scenarios. We are going to compare the ArrayList data structure that we created last week with LinkedList data structure that we created this week. Update your Excel table. It should look something like this one, but it should contain the actual times that you recorded last week for the ArrayList:  
     
   
2. Download the ClientData.csv file and place the file in your “Week 2 Project” folder. We will read the data in this file so we can work with a real-world sized dataset to test our algorithm.
3. Copy the Node.py file and the LinkedList.py files to this folder. We will use the LinkedList code to check the speed of this algorithm.
4. Copy your Client class to this folder.
5. In this same folder, create a file called LinkedListActualSpeed.py. Type your name and the current date at the top of the code. Then, import the LinkedList class, the Client class, the date module from the datetime library, the time module, the random module, and the sys module.
6. In the same LinkedListActualSpeed.py file, display your name and the current date in the output to show that you are the author of this code.
7. In the same LinkedListActualSpeed.py file, create a list and read the records from the ClientData.csv file into Client objects and place the Client objects into the list.
8. In the same Python code file, create an LinkedList object. Let’s test the first scenario. If you accept jobs at the back of the line and then process the jobs from the front of the line, which data structure is best? Write code to test how much time that it takes to add the Client objects from the list into the back of the LinkedList data structure. Add this time to the spreadsheet into the “Add many records to end of data structure” row in the Excel table.
9. Continue the LinkedListActualSpeed code and write code to test how much time it takes to remove all of the Client objects from the front of the LinkedList, one object at a time. Add this time to the spreadsheet into the “Pull all records off front of data structure” row in the Excel table.
10. Now, let’s test the second scenario. A customer service center receives calls from random customers. The customer service center must be able to quickly pull up the customer’s record. Is the LinkedList good for this scenario? Let’s see how long it takes to pull up random records. Write code to add all of the customers to the LinkedList again since we deleted all of the customers on the last step. Now, write code to test how much time it takes to display many random records. Add this time to the spreadsheet into the “Display random records” row in the Excel table.
11. Let’s consider the third scenario. Consider a credit card call center. The representatives may sign up new credit card accounts (add records). The representatives may answer customers’ questions (display random records). Finally, the representatives may delete a paid-off account at the request of the customer (remove records). Write code to test how much time it takes to add the Client records to the LinkedList, then randomly display records, and then randomly delete records.

# Deliverables Part 2

* Complete the Module 2 Course Project Presentation deliverable
  + LinkedListActualSpeed.py code
  + Screenshot showing the code running with your name and date in the output
  + Updated Excel “Table of Speeds” table