# CSC 236 T12 Exploring Linked Lists

## This is a team assignment designed as an in-class activity.

---

Note that this team assignment was created by Dr. Jan Pearce of Berea College

---

## Adapted POGIL Working Groups

You are to work in groups of three or four. There are typically 4 roles in this kind of activity:

a) **Facilitator** - Keeps the team on-task and on schedule.
b) **Quality Control Engineer** - Works to ensure all responses are high-quality.
c) **Process Analyst** - Considers how the team is working together and suggests ways in which it might improve.
d) **Reader/Spokesperson** - Reads the document aloud and speaks for the entire team.

See Adapted POGIL Roles for more details on these rolls.

If you have three people, combine the **facilitator and process analyst** roles. **These roles are designed to maximize communication, so we encourage you to TALK BEFORE writing,** except when you are explicitly told you may divide and conquer.

## Directions for use:

- **Note that teamworks are designed to be completed WITHOUT AI, and using AI on teamworks is prohibited because AI use will impede learning.**
- One person is to make a single copy of this worksheet for the entire group in their Google account.
- That person should share it with everyone in the group so they can all edit it.
- Please note that for all teamwork, all team members must be full participants in all of the work. For all teamwork that spans multiple days, if a member is missing on the second day of the teamwork, a copy should be made by the team so the work can be carried out separately by the smaller team using the copy, and by the missing team member using the original document.
- The reader/spokesperson changes the document title so that it is *csc236_t12_yourusernames*

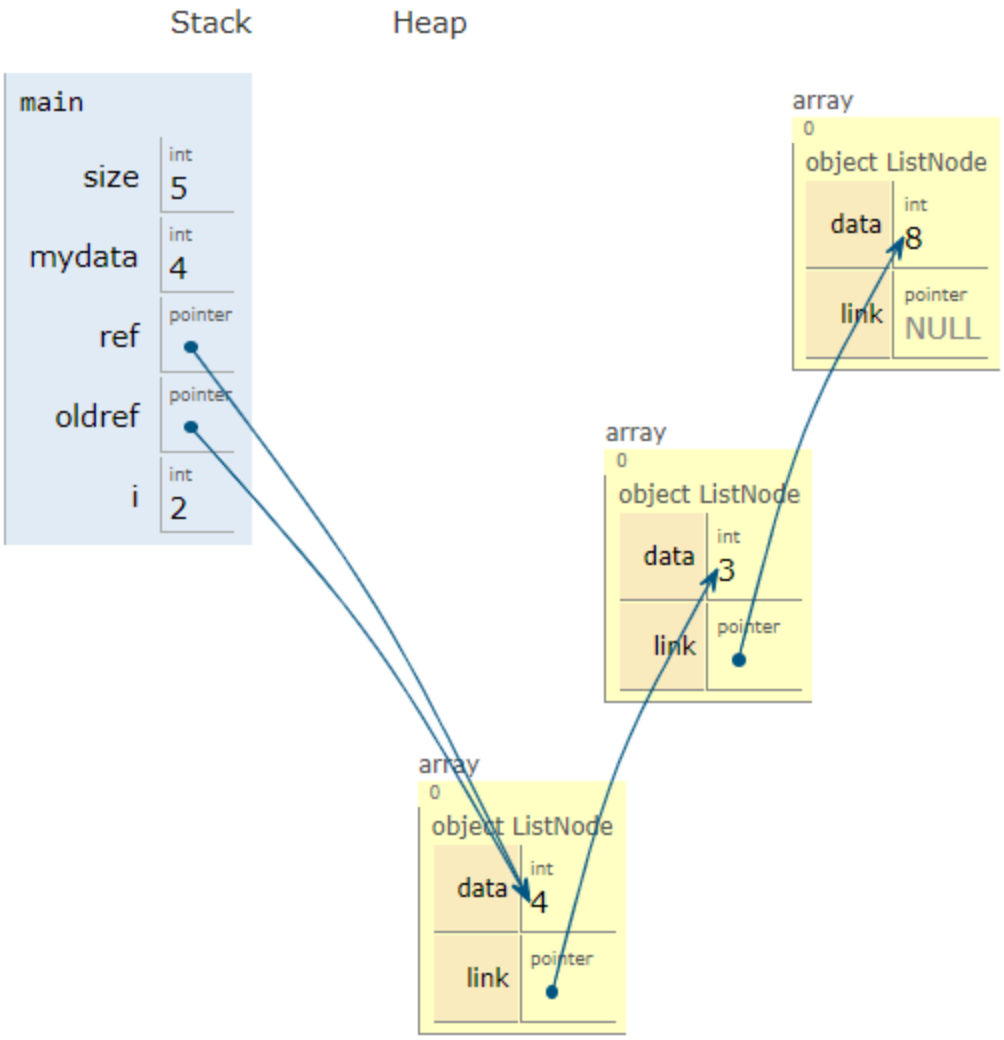
Pick a role **YOU HAVE NOT HAD A CHANCE TO DO YET**:

| Team Roles | Member Name |
|---|---|
| **Facilitator** | |

| | |
|---|---|
| **Quality Control Engineer** | |
| **Process Analyst** | |
| **Reader/Spokesperson** | |
| **TEAM NAME** | |

## Understanding Linked Lists

Consider the following image taken after the code has run for a while:




Use C++ Linked List Code to step through the code several times. Note that to rerun it, you need to click on the edit code button, change to the C++ language, and then push the visualize button.

1. (3 min) Are the same number of ListNodes created each time you run the program? Explain.

| |
|---|
| |

2. Describe the order that the nodes are appearing in the linked list by thinking about the need to access values starting at the head pointer. Is this list created last-in-first-out or first-in-first-out?

Explain.

Move back and forth in the sequence of steps in the code in the program to see the sequence of steps in memory which happen to create each new `ListNode` object.

3. (3 min) Describe this sequence of steps in memory and then explain why this makes sense.

4. (2 min) What difference from all of the other nodes, do you notice about the very last `ListNode` object in the linked list?

5. (3 min) If you did not know in advance how many nodes were in a linked list, how might you use the information to be sure to stop when you reach the end of the linked list when using a code structure like a while loop?

Note that during the creation of the linked list, the for loop iterates downward from `i = size - 1` until reaching `i = 1`.

6. (2 min) Explain why this strategy might make more sense than iterating upward even though the variable `i` is not used in the loop body:

## Enhancing the program

You will take this opportunity to edit the program you were given by adding two getter (or accessor) methods in the `ListNode` class. While you are making your changes, make sure you credit the original author and clearly indicate what you are changing.

(15 minutes) Open the link C++ Linked List Code and add a getter function for each of the private member variable values. Remember, these functions should **return** the values and not print them to the screen.

Once you are done, you have some options. You can (1) copy the lines you changed, (2) create a permanent link to the entire program with your changes, or create a shortened link (the permanent link is very long) if you prefer.

7. Copy one of these to the space below:

## Iterating Through a Linked List, Part 1

You probably noticed in the similarities/differences question above that each `ListNode` object has NO name. Hence, to access any given node, one needs to start with the head pointer and then iterate through the linked list.

8. If one was going to use a loop, what specifically would be an appropriate stopping condition? I.e. Very specifically, how would you know when you are done iterating through the list?

9. (10 minutes) Write a pseudocode algorithm for visiting every `ListNode` object in the linked list. *Note: You will make this question much harder if you go looking for this on the Internet, so please do not do so--just use the abstraction of the image and write pseudocode for the algorithm. You can figure it out!*

## Iterating using an STL

Hopefully, you now understand the key ideas in linked lists. Are you happy to know that modern C++ has linked lists as part of the Standard Template Library? Go to Forward List Example and try it out!

10. (10 minutes) Describe what this code seems to do.

## Iterating Through a Linked List, Part 2

Enhance the C++ Linked List Code so that **AFTER the entire linked list has already been formed,** you write a new loop in `main()` to iterate through the entire linked list and print out each of the data values in order. Of course, you can use your new accessor methods to do this!

> *Note that you could easily put a `cout` statement into the existing loop to print the value of the data attribute of each node when it is accessed, but that is **NOT** what we are asking you to do.*

Once you are done, you have some options. You can (1) copy the lines you changed, (2) create a permanent link to the entire program with your changes, or create a shortened link (the permanent link is very long) if you prefer.

11. Copy one of these to the space below:

| |
|---|
| |

## Summary

12. Complete the following table individually on the most significant or surprising thing learned in this activity by each team member:

| Member Name | Most Significant Thing Learned |
|---|---|
| | |
| | |
| | |
| | |

13. Please offer any suggestions for improvement of this activity from the team:

| |
|---|
| |

---

## Submission and Integrity

- Before submission, please ensure that all names at the top fully participated in the entire teamwork. Please note that if a teammate was absent for part of the teamwork, they must not be given credit for work they did not complete. If a team member was not present for part of the work, this must be explained at the top, and your team will have completed the work in a copy. If you forgot, you can make a copy now, and then undo all of the changes in the original back to the appropriate point for the missing teammate to complete.
- All team members must download a copy of this document as a PDF and upload it to Moodle.
- Note that by uploading this document to Moodle, you are asserting that this is fully the work of the team members listed at the top, and that you have not used any AI tools except those built into the IDE.