

Lab 4.3 – Implement an integer stack using a linked list.**Learning Goals**

- 1) Develop your understanding of linked lists.
- 2) Develop your ability to create object like functionality in C using a combination of structs, typedef, and pointers.
- 3) Develop your ability to use malloc to request memory on the heap.

Background

We previously implemented a stack with an array. In this assignment, we will implement a stack using a linked list.

For a stack, using a linked list is easier to implement than using an array. There are no special cases and you don't have to worry about running out of room.

Your Task

- 1) Create a new directory called **lab3**. The files in this assignment have the same names as files that we previously worked with in class.
- 2) Copy **stack.h**, **stack.c**, and **stack_test.c** from Schoology into that directory.
- 3) Complete implementing **stack.c**. The structs are already defined. (The node definition is exactly like the one we used in class for IntList.)

Where is the top of the stack? The simple trick is that the **head** points to the top element of the stack. So, for a **push**, the new element becomes the first element in the list. For a **pop**, the element pointed to by head is popped.

Make sure you can draw push and pop before you try to code them!

- 4) Once you are done with your implementation, run **stack_test.c** to check your results.