1. OLD   
   Introduction to our lab
   1. What is our prompt
   2. What is the purpose of our class
2. Populate Both Structures w/ push
   1. Add 50 to each, top every 10 ints to show where they’re at
   2. Visual Debugger side by side
3. Pop with top every 10 ints until we get the overflow exception

Prompt for pushes, prompt for pops, have be called  
  
December 10th Update

Need New Demo Program  
 We start with ArrayOfStacks  
Don’t go through the debugger, we can just show how the original one works  
Explain dilemma & reason why we need the double linked list

Go through each class and describe its parts/function/utility  
We’re going to need to push values to the stack, ints for simplicity, we could do this randomly or do this in order.

Random might be better so the fact that the items are different from the index is clear   
We’re going to need to demonstrate the popAt function and showcase 2 or 3 different interactions

1. Functionality of the popAt function
   1. How the accumulator helps us determine if an access stack contains the desired index
2. What happens with multiple popAts
   1. How the
3. What happens when an access stack is emptied
   1. Show bottom stack being emptied and what happens when middle stack is emptied
   2. Top too?

Create ppt

~~Upload Doc~~

~~Refine ArrayOfStacksDemo~~

~~Go Through DLL class with comments and annotate each case or step~~Refine Demo of DLL

“Here is our premise”  
“This is our first attempt”  
“Then we realized we’d need popAt”  
“We needed to figure out how to access the bottom of an inner stack”  
“If we only needed to do that once we could just use modulus, but that would rely on every stack except the top one being full at 10(capacity). We can achieve that by shifting all the elements down one each time an element is removed, or we can use a DLL and not worry about keeping all of the lower stacks at full capacity”

“PopAt presented this challenge: we need to access an element other than what’s at the top of the stack“ “What happens after we remove an element at a specific index? The items in the AccessStack shift down” “**Using a DLL allows us to only have to shift down the affected inner Access stack instead of having to shift everything above the index we removed**”  
“A DLL allows us to avoid manipulating all the elements above the removed index and still have the ability to access whatever index we need”

“We can access the top or the bottom of a DLL, and access the bottom is where we start when searching for an element with popAt ”  
“We used DLL to overcome this challenge”

* Using the DLL allows us to not have a variable access stack size & not consolidate