

Lab15: Stack Class

Note: **We are not using our 101 element array implementation of a Stack** in this project, read the directions closely and do not copy your old lab code!

The maximum number of elements: The implementation of the Stack class should allow for any number of elements, not capped at 100. To do this, follow this thought process:

- Start by keeping a 100 element int storage array to hold the Stack data and a separate int numberOfElements to keep track of where in that storage array you are.

- When the numberOfElements exceeds the length of the storage array, create a new int array 100 elements bigger than the previous, copy over the old data, and make that the new storage array.

Example,

```
Stack s = new Stack();
for(int i=1; i<=100; i++)
    s.push(i);
// Now, the Stack is full. If you push 1 more element, it would be out of bounds.
// But, instead of letting the program crash, make a new 200 element array, copy over
// the old 100 values, then add in the 101st element.
// Use the functions from Part 1 when you are making the stack bigger.
// You need to determine when the appropriate time to resize your stack is.
s.push(101);
```

Part 1: Create the following helper functions for your Stack class and use where appropriate:

```
// a. Create a separate copy of a. Do not just return a new pointer to the same array!
public static int[] copyArray(int[] a);
```

```
// b. Return a new array with all elements of a and b. Ex, append({1,2,3},{4,5}) => {1,2,3,4,5}
public static int[] append(int[] a, int[] b);
```

Part 2: Create a Stack class that represents a stack of ints. Include the instance variables:

```
private int[] storage;
private int numberOfElements;
```

Part 3: Include 1 constructor, the empty constructor.

Lab15: Stack Class

Part 4: Include getters for storage and numberOfElements, but do not create setters because **we do not want other programmers to be able to mess with the data by any means other than push and pop**. Reread the bold part. Most students will lose points here.

Part 5: Create push() and pop() methods.

Part 6: Create a toString() method that will return a String that looks like this: "[1, 2, 3]". It should only contain the portion of the array that is the stack.