# Lab Assignment 20 (Stacks, Lists, & Queues)

**NOTE: Your program should be your own work and not a copy of anyone else’s code. Copied code will not earn any points.**

1. Write a program named **Lab20A** that will do the following:
   * Create a void method that receives an ArrayList of integers as a parameter and will:
     1. Print the received ArrayList with a label stating that it is the original list
     2. Create a new ArrayList named **list1** that is a copy of the received ArrayList  
        (The *clone* method from the video sample code could be a good way to do this. This way you will not change the original ArrayList.)
     3. Create a new ArrayList named **list2** that contains (0,10,20,30,40,50,60,70,80,90,100)

Use this as an example (and add it to your notes):

ArrayList<Integer> listName = new ArrayList<Integer> (Arrays.asList(0,10,20,30,40,50,60,70,80,90,100));

* + 1. Use the *contains* method to see if **list1** contains 45 and print a statement to say if it does or not.
    2. Use the *removeAll* method to remove all the items from **list1** that are in **list2**
    3. Print **list1** with a label to say that the items from **list2** have been removed
  + Create an int method that receives an ArrayList of integers as a parameter and will:
    1. Use an iterator object to go through the ArrayList and total the values
    2. Calculate and return the average of the values
  + In the main method create an ArrayList of integers and fill it with 20 random numbers between 0 and 100. Call both methods sending the ArrayList as a parameter and print the value returned by the second method.

Then sort the ArrayList (using Collections.sort) and print it.

1. Create a program named **Lab20B**.
   * Create a secondary class named **Book** with:
     1. Instance variables:
        1. A String variable for the book title
        2. A String variable for the book’s author
        3. An int variable for the number of pages in the book
     2. A constructor that accepts 3 parameters to fill in the instance variables. (The parameter data types should match the instance variable types.)
     3. An int method that returns the number of pages (no parameters)
     4. A String method that returns the title (no parameters)
     5. A String method named **toString** that returns a string with each instance variable value and a label in front of each.
   * In your main Lab20B class write a void method that receives a stack of **Book** objects as a parameter. It should:
     1. Print the number of objects in the stack
     2. Print the statement, “Books Read in Challenge: “
     3. Let’s imagine that we have joined a challenge to read 800 pages this month.   
        This method will take one book from the top of the stack at a time and keep track of the total number of pages read. It should stop when the total is >= 800 pages, and the other books will remain in the stack.
     4. For each book we “read”, use the *pop* method to remove the top book from the stack; then print its information and add its number of pages to the total variable. (Remember that stacks use Last In – First Out ordering, so items popped from the stack should be in reverse order of how they were added.)
     5. After the loop has finished, print the total number of pages read and a blank line after it.
     6. Note: I will test your code with a text file with different books in it, so don’t just count how many books make 800 pages and loop that many times.
   * In the main method:
     1. Create a stack of **Book** objects
     2. Read lines from the text file, **Lab20B.txt** to get the data for your **Book** objects. Use the push command to add each object to your stack.

**NOTE:** The title & author values have more than one word, so you need to use the nextLine() method to read each of them from the text file. On the number of pages line, you only need to get the integer value and then make the program go to the next line. Do NOT change the data in the input file because I will be testing your program with the original input file. (If you need help with this, ask me.)

* + 1. Print the stack and a blank line after it
    2. Call your void method sending the stack as a parameter.
    3. Print the current stack

1. Create a program named **Lab20C** that will work with a Queue data structure. We will simulate a line of customers who want to buy widgets from our company.
   * Create a second class named **Orders** that will have the following:
     1. Two instance variables holding the customer’s name (String) and the number of widgets being ordered (int)
     2. A constructor that receives 2 parameters which will be used to fill in the instance variables.
     3. An int method that will return the number of widgets ordered (no parameters)
     4. A toString method that returns a string with both instance variables and labels (no parameters)
   * Back in the main class
     1. Create a void method that will receive an **int** “available widgets” number and the queue of orders as parameters. Remove each order in the queue until there are not enough available widgets to fill the orders still in the queue.

(Use peek to check if there are enough available widgets left for an order before you remove it from the queue.)

* + - 1. Print the statement: “Orders Processed: “ at the beginning of the method before you start processing orders.
      2. For each order removed, print its information (using toString shortcut)
      3. After the loop has finished, print the number of available widgets left
      4. Print the remaining queue (with a label)
    1. In the main method:
       1. Create a queue of **Order** objects
       2. Read lines from the text file, **Lab20C.txt** to get the data for your **Order** objects. Add each object to your queue.
       3. Print the queue and a blank line after it
       4. Call your void method sending the value 45 and the queue as parameters.
       5. Print the current queue.