**CIS163**

**Clean things up for the Final Exam**

**Objectives** (after completing the lab you will be able to do:)

* Handle UML on the Test
* Handle Polymorphisms on the Test
* Handle Exception on the Test
* Handle Testing on the Test
* Handle Recursion on the Test
* Handle References on the Test
* Handle Single Linked List Test
* Handle Double Linked List Test
* Handle Streams and Lamda Test
* Handle Stack and Queues Test

**Activities:**

1. ***Am I OK with UML?***

Try this:

* Goto page: 254 of your book and attempt to create a UML diagram from the program found there, specifically the LeftRight.class
* Goto page 385 an attempt to write “skeleton code” from the UML diagram found there

(remember to change the arrows to fit CS standards)

**If you are having issues, please see the instructor (or lab assistant) for help**

1. ***Am I OK with Polymorphism?***

Try this:

Start with the previous lab on Polymorphism.

* For a warm up, attempt to remove the cast statements (awardBonus, addHours) in the Staff class.
  + **When you done with this, please change the code back to require a cast statement**
* For another warm up, attempt to create a CEOExecutive that inherits SuperExecutive
  + Add this new CEOExecutive into the staffList array in the last position
  + Create all needed methods (toString, constructors, etc.) and make up data as needed.
* Change the code in the Staff constructor, specifically, the two cast statement lines:

((Executive)staffList[0]).awardBonus (500.00);

((Hourly)staffList[3]).addHours (40);

so that the lines 19 – 33 can be in any order. In other words, you will not know stafflist[0] is an executive and staffList[3] is an hourly. They could be in any order.

**If you are having issues, please see the instructor (or lab assistant) for help**

1. ***Am I OK with Exceptions?***

Try this:

Start by downloading/installing the MExceptionMulti Class found on BB

* Run the program and predict the output
* Change the -10 for the constructor (Arraylist) at line 12 to the value of 10
* Run the program and predict the output
* Change line 12 to ArrayList l = **null**;
* Run the program and predict the output
* Create a checked exception named (myCheckException) and throw this new exceptions at line 20, catch that exception in the main program. Do checked exception make sense?

**If you are having issues, please see the instructor (or lab assistant) for help**

1. ***Am I OK with Junit testing?***

try this:

Start by downloading/installing the Lock Class found on BB (week 1)

* Create a JUnit class named TestPrep, and then create JUnit test methods that:
  + Checks if the Lock opens using the isLock() method
  + Checks if an IllegalArgumentException is throw with new Lock(“”);
  + Checks for two locks being equal.
  + Checks for master code working using the isLock method.
  + Now, run your Junit test class and see if the lock class passes.

**If you are having issues, please see the instructor (or lab assistant) for help**

1. ***Am I OK with Recursion?***

Try this:

Open a browser and goto to: http://codingbat.com/java/Recursion-1

* Attempt to solve the following:
  + Factorial, Count7, Fibonacci, CountHi, pairStar

**If you are having issues, please see the instructor (or lab assistant) for help**

1. ***Am I OK with References?***

Try this:

Start by downloading/installing the Example and Simplelass found on BB (week 1)

* Run the program and predict the output
* Uncomment the block if lines starting at (about) line 18
* Run the program and predict the output
* Comment lines 8 and 9 (i.e., s1.inc; s2.inc()….)
* Run the program and predict the output

**If you are having issues, please see the instructor (or lab assistant) for help**

1. ***Am I OK with Single Linked list (with or without a tail)?***

Try this:

Start by downloading/installing the Linked list code we did in lab (Node class on BB, Documents)

* Write the following methods.
* public void insertBefore (int index, String data) {
* public void insertAfter(int index, String data) {
* public String removeBottom () {

**If you are having issues, please see the instructor (or lab assistant) for help**

1. ***Am I OK with Double Linked list (with or without a tail)?***

Try this:

Start by downloading/installing the Linked list code we did in lab (DNode class on BB, documents)

* Write the following methods.
* public void insertBefore (int index, String data) {
* public void insertAfter(int index, String data) {
* public String removeBottom () {

**If you are having issues, please see the instructor (or lab assistant) for help**

1. ***Am I OK with Streams and Lamdas?***

Try this:

Start by downloading/installing the lamdas/stream code we did in lab

* Given the code below, write a Lamda function to sort by last name.
* Given the code below, write a Lamda function to sort by last name, then…;
  + If same last name, use first name.
* Given the code below, write a Stream function to filter by last name that contains a “John”.
* Given the code below, write a stream to sort by last name.

// Other code maybe needed, for example: is the compareTo needed?

ArrayList<Student> friends = **new** ArrayList<Student>();  
  
 friends.add(**new** Student (**"John"**, **"Smith"**, **"610-555-7384"**));  
 friends.add(**new** Student (**"Sarah"**, **"Barnes"**, **"610-555-7384"**));  
  
  
 **If you are having issues, please see the instructor (or lab assistant) for help**

1. ***Am I OK with Stack and Queues?***

Try this:

Start by downloading/installing the stacks and Queue code found on BB (documents)

* Write a Stack Class (i.e., pop, push, isEmpty) using an Arraylist, Single linked list
* Write a Queue Class (i.e., deQue, enQue, isEmpty) using an Arraylist, Double linked list

**If you are having issues, please see the instructor (or lab assistant) for help**

1. ***Am I OK with ALL THE REST of the stuff we did in CIS163.***

Try this:

**Look at BB in the Document section and see if all the code there makes sense.**

**Example 1, there is code on BIG O in Documents, do you understand Big O?**

**Example 2, there is code on JUnits in Documents, do you understand Junits?**

**Example 3, there is code on Polymorphisms in Documents, do you understand Polymorphism?**

**Example4, Numbers 5 9 2 3 7 1 0 4 6 Sort using: Selection, Insertion, Quick, Merge.**

**Etc…**

**If you are having issues, please see the instructor (or lab assistant) for help**