COMP232 – Data Structures & Problem Solving
Fall 2023

Homework #1
Recursion and Backtracking
[Total points: 100]

1. [5 points] Confirm that you have properly cloned the **hw-01** project by running the `RunMe` class (Click the Run Icon in the Eclipse toolbar, or right click on the class name in the "Package Explorer", point at "Run as" and choose "Java Application"). What output is generated in the "Console Tab"?

2. [15 points] Open the CountXWithDubs class and complete the countXWithDubs method. Your solution must be recursive. Complete the method in Eclipse. When you have completed all parts of this homework, push your code to GitHub.

3. [10 points] Open the NestedParentheses class and complete the validNesting method. Your solution must be recursive. Be sure to test your solution by running the class as a JUnit Test. Complete the method in Eclipse. When you have completed all parts of this homework, push your code to GitHub.

4. [10 points] In class we drew the left part of the recursive call tree for the subset method when invoked on "ABC". Draw, very neatly, the right part of the recursive subset tree when invoked on "ABC". Use the same notation that we used in class. Answer this question here. When you have completed all parts of this homework, save this document as PDF by creating a folder named "noncode-answers" in your git repository for this homework assignment, then push your code to GitHub.

5. [20 points] Open the Combinations class and complete the private combinations method. Hint: Develop a solution by modifying the subset method we studied in class. Be sure to test your solution by running the class as a JUnit Test. Once you get it working try to modify the code further to reduce the number of recursive calls made during a run. It is possible to compute combinations("ABCD",2) with 19 calls and combinations("ABCDE",3) with 34 calls. You can edit the main method and run Combinations as a Java Application to see the number of calls made. Complete the method in Eclipse. When you have completed all parts of this homework, push your code to GitHub.

6. [10 points] Run the SplitArrayPrintSolution class as a Java Application. What output is generated? Why? Answer this question here. When you have completed all parts of this homework, save this document as PDF in the noncode-answers folder of your git repository for this homework assignment, then push your code to GitHub.

7. [10 points] Open the SplitArrayPrintSolution class. The private the splitArray method in this class is nearly identical to the one we studied in class. The only change is that it has had 2 parameters (gp1List and gp2List) added to its signature. These parameters provide a means for the private splitArray method to keep track of the actual values in each group. The values added to these lists are then printed by the public splitArray method when a solution is found. Currently the private splitArray method does not add any values to these lists. Modify the private splitArray method such that the solution is printed correctly. Hint: It is only necessary to add 4 lines of code to the private splitArray method. Complete the method in Eclipse. When you have completed all parts of this homework, push your code to GitHub.

8. [20 points] Open the SplitOdd10 class and complete the splitOdd10 method. Your solution must use recursive backtracking. Be sure to test your solution by running the class as a JUnit Test. Complete the method in Eclipse. When you have completed all parts of this homework, push your code to GitHub.