**Project 4**

**Ch: 18 | Recursion**

**Lessons Learned:**

In order to print out all the permutations of a string I made use of both a loop and a recursive call in the function. The function is comprised of two methods, one of them is is used in the other method which is called to get the result.

Inside the loop of the method, the function makes a recursive call to itself with each iteration, and it eventually passes the complete permutation to the first argument and prints it out the permutation to the console when the second argument reduces to an empty string, in other words when there is no more possibility left to place it for that particular fixed part of the string (which is stored in the first argument).

Making use of substring is an essential part of the permutation function, since it allows to change the arrangement of the characters in the fixed part of the string (in the first argument) each time after the previous permutation is printed out.

With this exercise, I saw how permutations are utilized when we need to use nested loops in a very complicated way. Permutations make out code more concise and make the logic clearer.

However, in order to use them appropriately, I understand that I need to do more exercises with it since the algorithms that needs recursion is not very easy to implement.

The input string is “abd”:

Graphical user interface, text

Description automatically generated

The input string is: “abcd’

Text

Description automatically generated

**Ch: 19 | Generics**

**Lessons Learned:**

In this chapter, I learned that generic classes and methods save us from writing code for various data types. With generics, we can write code that can be applied to different types. This also helps to prevent programmers from getting invalid type exceptions at the compile time.

In the max generic method, I made use of compareTo method since it accepts Object as a parameter in addition to String, and in ArrayList and we can only have reference types.

Finding the max value in an array list is a straightforward process and with the use of a for loop, one can achieve updating the max value with each iteration and return the max value after the loop at the end of the method.

**Screen Shot:**

Text

Description automatically generated

**Checklist**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** |  | **Y/N** | **Comments** |
|  | Source java files | **Y** |  |
|  | Files: |  |  |
|  | LastNameFirstinitial\_Project04.zip | **Y** |  |
|  | LastNameFirstInitial\_Project04.doc | **Y** |  |
|  | Program compiles | **Y** |  |
|  | Program runs | **Y** |  |
|  | Checklist is completed and included in the Documentation | **Y** |  |
|  | Documentation file: |  |  |
|  | ~~Comprehensive Test Plan~~ |  |  |
|  | Screenshots of running program | **Y** |  |
|  | ~~UML diagram~~ |  |  |
|  | Lessons Learned | **Y** |  |
|  | Checklist | **Y** |  |