**Kenneth Tillit**

**Due 04/11/2022**

**Searching & Sorting Project**

1) Describe auto-boxing, including why it is useful. Write a few lines of code that auto-box an int into an Integer, and un-box an Integer to an int.

Auto-boxing is the process of utilizing both wrapper class objects and primitive types throughout code, allowing for much cleaner looking code.

Text

Description automatically generated

2) Create a class called Person. The person class will have the following attributes: first name, last name, and phone number. Create a class called TestData. Declare an ArrayList of Strings. Add the names to the ArrayList. Output the Person objects from the ArrayList into the console using the enhanced for-loop.

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

3) Sort the list using the method Collections.sort. Output the sorted List. Shuffle the list, and output the shuffled list. Note that Collections (with an s) is a class, while Collection is an interface. The Collections class has many useful static methods for processing interfaces, including the sort method.

Text, letter

Description automatically generated

Text, letter

Description automatically generated

4) Describe why an equals method and a compareTo method are required to achieve searching and sorting of the elements of a list.

An equals method would be used in the searching part of searching & sorting, searching through the list to find something that is approximate to what is being searched for, hence “equals”. As an example, searching for patientID “18” will pull up the patient with ID 18.

A compareTo method on the other hand is used for the sorting, comparing parts of the list with other parts to determine the order in which they are aligned, as according to what the user is trying to sort the list into. For example, if someone is trying to sort in descending order, then compare to would compare the numbers 8 and 17 to see which one is higher, and place that accordingly, in this case, before the number 8.