Lab Assignment #2

Due Date: **On or before Second Class of Week 4** **Marks/Weightage: 30/10%**

Purpose: The purpose of this Lab assignment is to:

- Practice the use of generic methods and classes, extension methods, custom data structures such as Linked Lists, Stack and Queues

References: Read the lecture notes/ppts and code examples. This material provides the necessary information that you need to complete the exercises.

Instructions: Be sure to read the following general instructions carefully:

This lab should be completed individually by all the students. You will have to demonstrate your solution in a scheduled lab session and submitting the assignment **through drop box link on e-Centennial**.

>> At the start, you must name your **Visual Studio 2019 solution name** according to the following rule:

FirstName-LastName_SectionNumber_COMP212_Labnumber

For Example: ***John-Smith_Sec003_COMP212_Lab02 (say if your section number is 003)***

>> And after that your **project name** should be as follows:

FirstName-LastName_SectionNumber _Labnumber

For Example: ***John-Smith_Sec003_Lab02***

>>Each exercise should be placed in a separate package named as firstname_last-name_*exercise1*, firstname_last-name_*exercise2* etc.

>> After you complete, exit eclipse and go to workspace folder, zip it up and you will get the following zip file.

FirstName_LastName_SectionNumber_COMP212_Labnumber.zip

Example: **John_Smith_Sec003_COMP212_Lab02.zip** *(if your section is 003..)*

>> Apply the naming conventions for variables, methods, classes, and packages:

- *variable names* start with a *lowercase* character for the first word and uppercase for every other word
- *classes* start with an *uppercase* character of every word
- **namespace** use only *lowercase* characters
- *methods* start with a *uppercase* character for the first word and uppercase for every other word

Note: Late submissions are accepted until up to three days past due date with 25% deductions. After that no submission will be considered.

Exercise 01 *[5 marks]*

Write a generic method, ***Search*** [int Search<T>(T [ ] dataArray, T searchKey)] that searches an array using linear search algorithm. Method ***Search*** should compare the search key with each elements in the array until the search key is found or until the end of the array is reached. Take the size of array as 10.
If the search key is found, return its location in the array; otherwise return -1.
Test this method by passing two types of arrays to it. (an integer array and a char array)

Display the generated values so that the user knows what values he/she can search for.

[Hint: use (T: IComparable<t>) in the where clause for method Search so that you can use method CompareTo() to compare the search key to the elements in the array]

Exercise 02 *[5 marks]*

Implement an extension method for built-in class ***StringBuilder*** to count the number of words contained in a ***StringBuilder*** object.
For example, if a ***StringBuilder*** object ***sb***="This is to test whether the extension method count can return a right answer or not", the number of words contained in ***sb*** is 16.

Exercise 03: *[20 marks]*

You need to enhance your generic **LinkedListLibrary<T>** (.dll file) class library project (which you have completed in the class/Lab) , by adding the following methods apart from the existing methods:

- T Minimum<T>() method which will return the smallest item/node value
- T GetLastNode() which will return the last element in the linked list

[Note: Create a new solution, and to that solution add a new class library project. Enhance it as per requirements. After that add another project- LinkedListLibraryTest to the above solution where you can test the above library by adding a reference to test project. Take out the build of the above library in the Release Mode.]

Test this library by using it in the project – LinkedListLibraryTest by creating two linked lists of integers and doubles (containing at least 5 elements each) and calling the methods Minimum() and GetLastNode()

You need to enhance class library project **QueueInheritanceLibrary<T>** (generic version), which is derived from LinkedListLibrary<T>, by adding following method apart from the existing ones:

- T GetLast<T>() which will just return the last element in the queue and not delete it.

Test this library by using it in the project – **QueueInheritanceLibraryTest** by creating two linked lists based queue objects of integers and doubles and calling the method GetLast() and Minimum().

Evaluation: Functionality	
Correct implementation of classes (instance variable declarations, validations, constructors, properties class methods etc.)	70%
Correct implementation of test classes (declaring and creating objects, calling their methods, interacting with user, displaying results in use friendly way)	20%
Comments, correct naming of variables, methods, classes, etc. and exception handling	5%
User Friendly input/output	5%
Total	100%