Lab Assignment #5

Due Date: Mid-night (11.59 pm) Sunday 29th March, 2020 ***Marks/Weightage: 30/10%***

Purpose: The purpose of this Lab assignment is to:

- Practice the use of Asynchronous and Parallel Programming

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 eCentennial on or before the due date.

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_Lab05 (say if your section number is 003)

>> And your **project name** should be as follows: FirstName-LastName_SectionNumber _ExerciseNumber

For Example: John-Smith_Sec003_Exercise01

>> If your lab assignment has more than one exercise, then each subsequent exercise should be added to the same solution created above and named as firstname-last-name_exercise2, firstname_last-name_exercise3 etc.

>> After you complete, exit Visual Studio and go to solution folder, zip it up and you will get the following zip file. FirstName_LastName_SectionNumber_COMP212_Labnumber.zip For example: John_Smith_Sec003_COMP212_Lab05.zip (if your section is 003..)

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

- 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

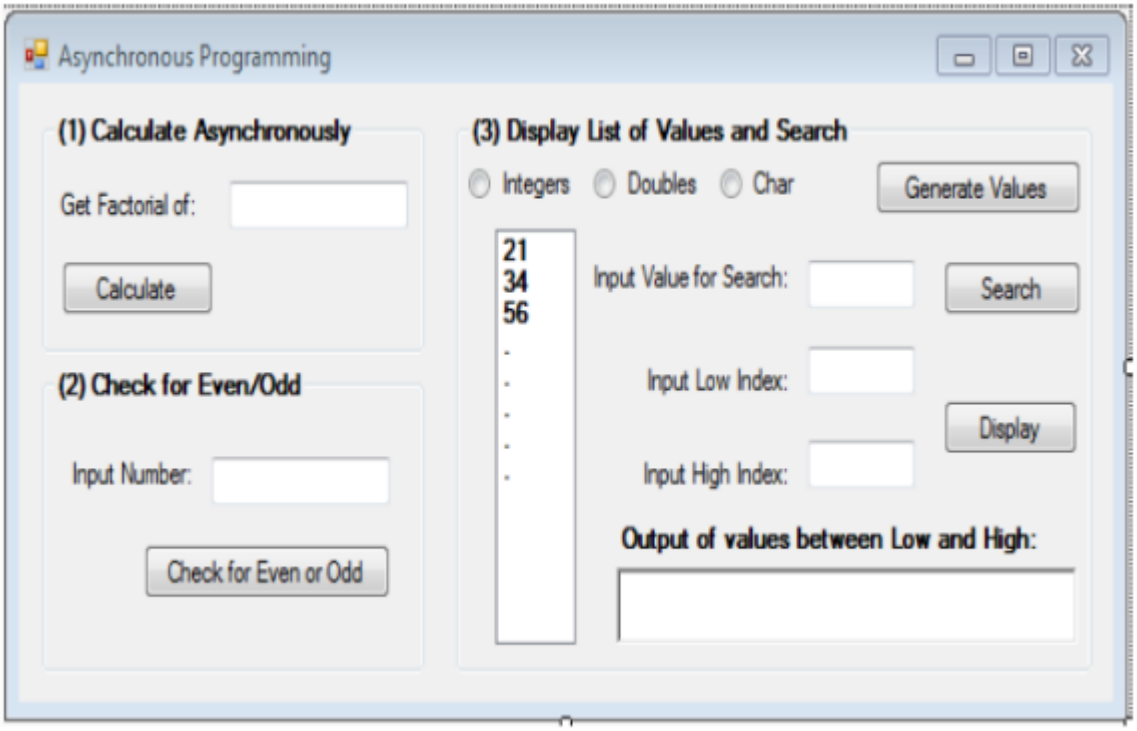
- namespace use only lowercase characters

- methods start with an 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. You must implement exception handling.

Exercise 01:

a) Create a **Windows Form App/WPF** to as shown in the screenshot. *[5 marks]*



b) For the first group box – Calculate Asynchronously, you would be creating a method –> long **Factorial** (long num) which is being called asynchronously (defined by the use of async, await and Task objects) in the event handler of **Calculate** button. You would be writing recursive version of Factorial. Also you need to validate all the input values and handling of all the possible exceptions. *[5 marks]*

c) For the second group box – Check for Even/Odd, you would be creating a two methods (by demonstrating the use of delegates and Lambda expression) - **IsEven** and **IsOdd** which takes an integer as input and return the true/false, and calling and using these methods in the event handler of **Check for Even or Odd** button. Also you need to validate all the input values and handling of all the possible exceptions. *[5 marks]*

d) For the third group box – Display List of Values and Search, implement the following: *[5 marks]*

i) **Generate Values** button should generate 10 values (between 10 and 99 and by making use of Random number generator) depending upon the selected radio button choice. Use list box control to display the generated values as shown in the screen shot.

ii) **Search** button should be able to search correctly an input value from the list of values in the listbox. Make use of Message Box to display the result. You need to define a generic search method – **SearchData** which takes a generic list<T> and value to be searched as inputs and returns boolean. You would be calling this method in the Search button event handler. Also you need to validate all the input values and handling of all the possible exceptions.

iii) **Display** button should display the range of values between the valid index values from the list of values in the listbox. You need to define a generic display method –