Lab Assignment #7 – Using Asynchronous Programming and Lambdas Ch20, C21, Ch23 (Sec001-002)

Student: ______________________

Due Date: **First Class of Week 13**

Marks/Weightage: 20/5%

Purpose: The purpose of this Lab assignment is to:

- Practice the use of Asynchronous Programming, Delegates and Lambdas

References: Read the course's textbook chapter 20 , 21 and Ch23 ppts, notes and class code examples (You can also refer previous chapters if you need to.) This material provides the necessary information you need to complete the exercises.

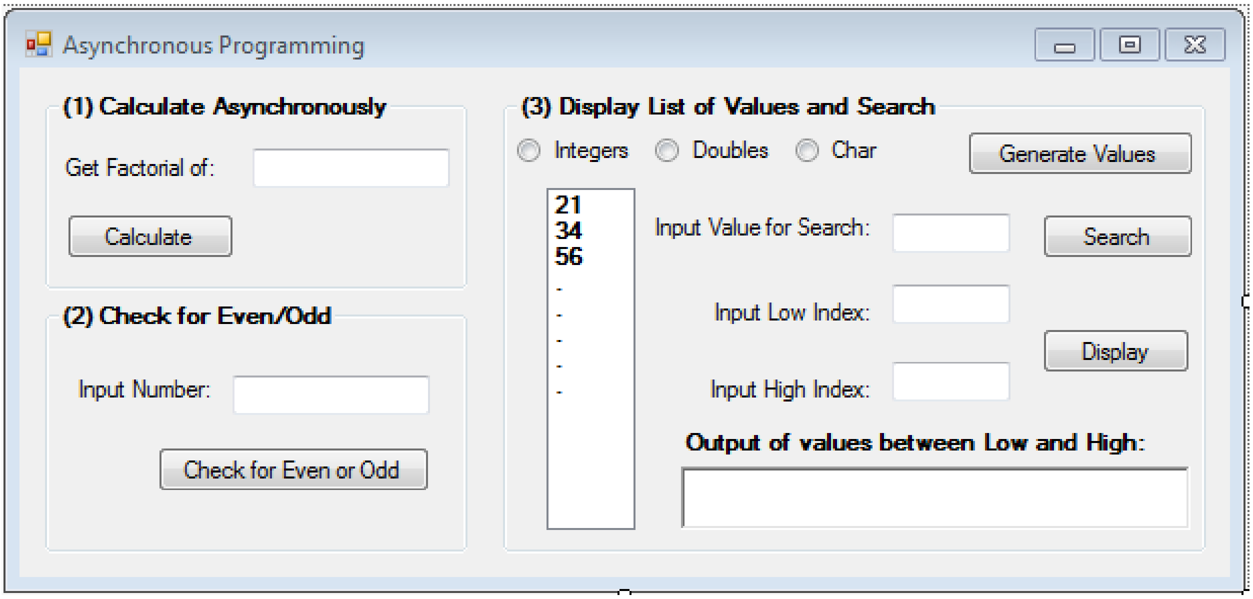
Instructions: Be sure to read the following general instructions carefully:

- This lab assignment should be completed individually by all the students.
- The solution folder **must be named** using the first name and last name followed by Lab assignment number and section number. For the student name - John Smith, the solution folder name should be *John-Smith_Lab07_Sec001*. Exercises in the lab assignment should be added as project(s). For example, exercise 01 name should be **John-Smith_Lab07_Ex01**, for the second exercise **John-Smith_Lab07_Ex02**, and so on**.**
- You will have to demonstrate your solution in a scheduled lab session and submitting the zipped solution/projects through the **Dropbox** link on **eCentennial**.
- You are required to follow the variable/control naming guidelines and must also implement exception handling in all the exercises.

Note (Very Important): *Late submission past due date is NOT allowed/accepted.*

Exercise #1:

a) Create the following Win form as shown below. ***[5 marks]***



21
34
56
-
-
-
-
-

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 – **PrintData** which takes a generic list<T> and low index value and high index value as inputs and returns a list of values between low and high index. You would be calling this method in the Display button event handler. Also you need to validate all the input values and handling of all the possible exceptions.