Lab Assignment #3

Due Date: **On or before Mid-night Sunday, 23rd Feb, 2020** **Marks/Weightage: 30/10%**

Purpose: The purpose of this Lab assignment is to:

- Practice the use of Windows Presentation Foundation, LINQ extension methods, built-in 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_Lab03 (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_Lab03***

>>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_Lab03.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

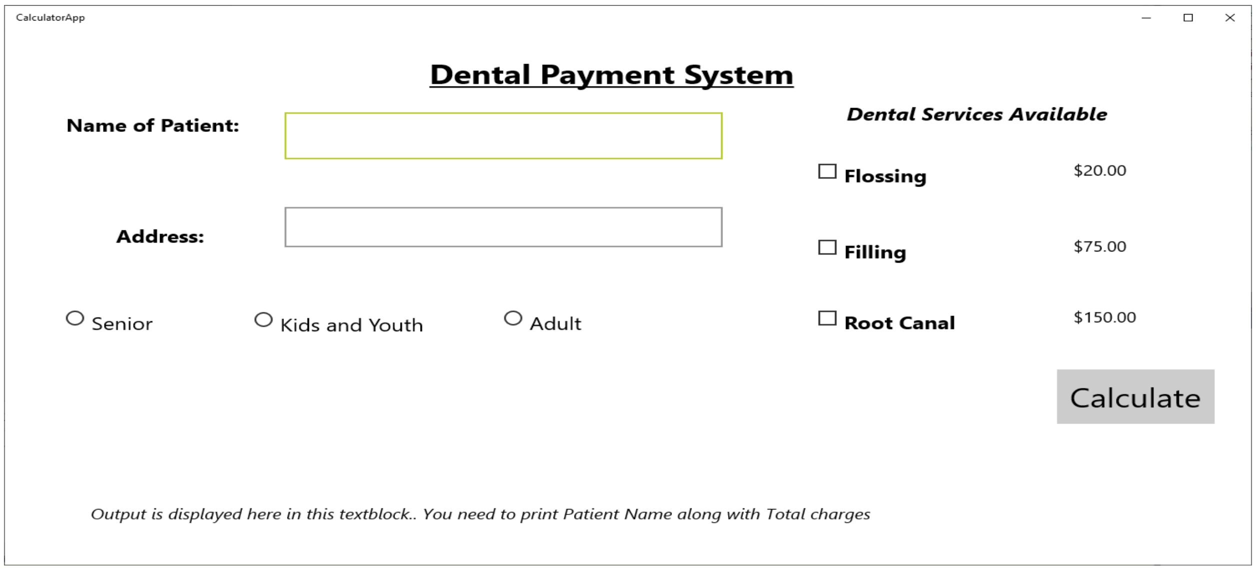
Note: Late submissions are accepted until up to three days past due date with 25% deductions. After that no submission will be considered.

Exercise 01

[15 marks]

Build the following Dental Payment System App using **WPF**. You can use appropriate layout controls.

If patient is Senior, then give 10% discount, if he/she is in category –Kids/Youth then 15% discount.
Add one combo box under Address textbox (drop down for Provinces – Alberta – HST 7%, Ontario- HST 13% and Quebec- HST 6%).
As per the selection of the province, Total Charges should be calculated accordingly.



Output is displayed here in this textblock.. You need to print Patient Name along with Total charges

Exercise 02: Based on LINQ extension methods.

[15 marks]

Create an Invoice class which includes four properties – a PartNumber (type int), a PartDescription (type string), a Quantity of item being purchased (type int) and a Price(type decimal).

Use the following sample data for Invoice class objects:

Part Number	Part Description	Quantity	Price
87	Electric Sander	7	57.98
24	Power Saw	18	99.99
7	Sledge Hammer	11	21.50
77	Hammer	76	11.99
39	Lawn Mower	3	79.50
68	Screw Driver	106	6.99
56	Jig saw	21	11.00

Perform the following queries on the array of Invoice objects and display the results:

a) Use LINQ to select from each Invoice the PartDescription and value of the Invoice (i.e. Quantity * Price). Name the calculated column as InvoiceTotal. Order the results by invoice value in ascending order. *[Hint: use let]*
b) Part description of the part who has highest quantity.
c) Average price of the parts.

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%