## CSIS1175-50 Assignment 4 Winter 2018

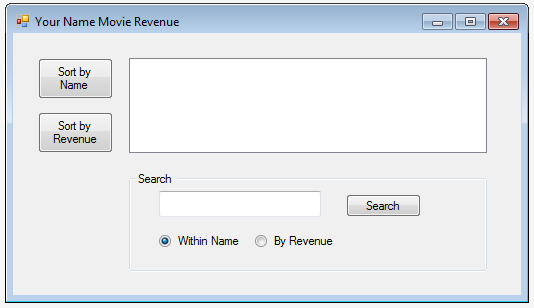
**YOU CAN DO THIS ASSIGNMENT WITH 1 PARTNER (NOT 20)**

**Due Date: Apr. 12th/2018**

**You can do this assignment with ONE partner**

**Create the following application**

This assignment will have the following GUI form below.

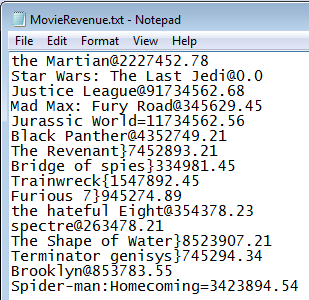


**Note:**

1. Use your **own judgement** on the sizes of the various GUI components
2. The font for the GUI components should all be default.
3. Please replace “Your Name” in  with your own name.
4. It is a **list box** that is next to the two sort butons.
5. The GUI component next to the search button is a **text box**

**Functionality**

Firstly, the input data for this assignment is from the **line sequential text file** called **MovieRevenue.txt**, with the content as shown below.

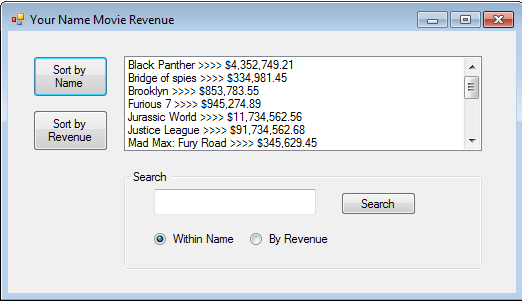


It shows the name of the movie and the revenue, however, the name and the revenue is separated by either ‘**{**’ or **‘}’ ‘@’** or ‘**=**’

You can assume that all **names and revenue** will be **unique** (i.e. no two movie title or revenue fiqures will be the same)

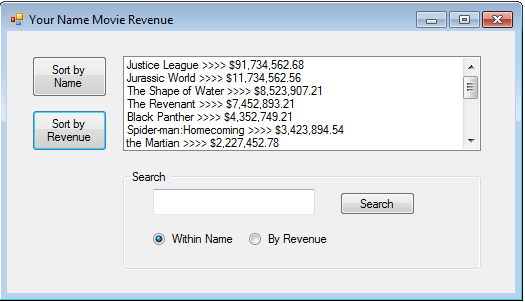
The following describes the functionality of the various GUI components.

When the user clicks the , it will display the movie titles by **name** in **ascending** order, **irrespective of case**, i.e:



**Note**, make sure to **clear the previous list box content** before displaying the new content.

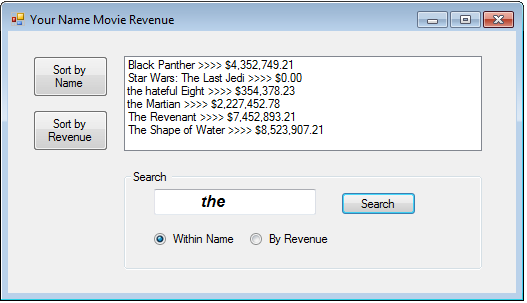
When the user clicks the , it will display the movie titles by **Revenue figures** in **descending** order i.e:



**Note**, make sure to **clear the previous list box content** before displaying the new content.

The following looks into the **search** part of the assignment.

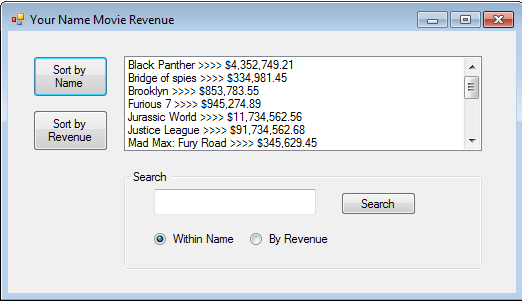
If the user clicks the  and enters  . Upon clicking the , this will search for all movie titles with “**the**” anywhere inside the movie title. The print screen below shows the output of this search.



Note the search should **NOT be case sensitive** and should **handle leading or trailing spaces**. Further, the output should **be *sorted in ascending sequence of movie title***.

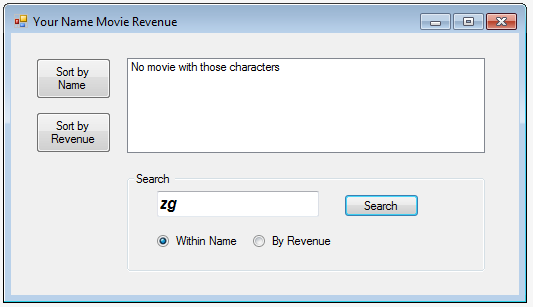
**Also**, make sure to **clear the previous list box content** before displaying the new content.

If the user **enters nothing or just spaces**, then upon clicking on “Search”, it should display all movie titles, i.e.:



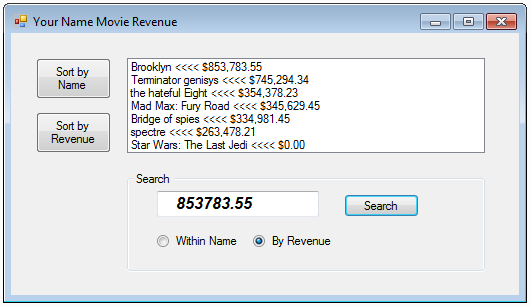
**Again**, make sure to **clear the previous list box content** before displaying the new content.

If there are **no movie titles with those characters in them**, then display with an appropriate message, i.e.:



**Also**, make sure to **clear the previous list box content** before displaying the new content.

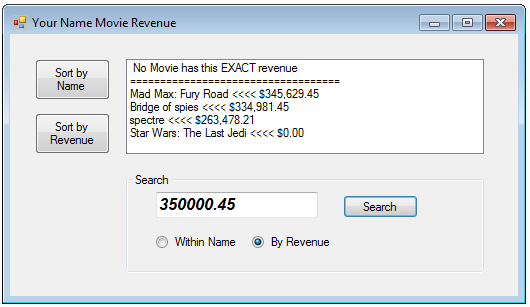
The user can also search by **revenue figures**. Note, it will **not only locate and display the movie title with that exact dollar amount**, but **also display all movie titles below that amount too**. The output will be sorted by ***descending*** order of revenue figures, i.e.:



(**Note also that the symbol to use is “<<<<”, rather than “>>>>”, when searching for specific numbers by revenue**)

As usual, make sure to handle **leading and trailing spaces and clear the previous list box content** before displaying the new content.

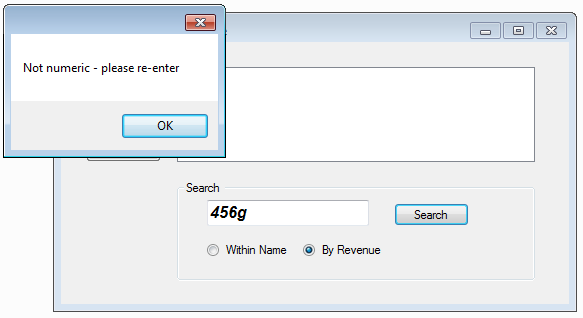
If the user enters an amount that no movie title has, then display **all the movie titles with an amount that is below what is entered.** Also display a message at the top to indicate no title had that exact amount you were searching for, i.e.



(**Note also that the symbol to use is “<<<<”, rather than “>>>>”**)

**Also**, make sure to **clear the previous list box content** before displaying the new content.

Finally, if the user entered any non numeric text into the search box while searching by revenue, then display an error box, i.e.:



The **above applies** if the user entered **nothing or just spaces** while search for revenue figures.

**Methods required**

Besides the methods for the GUI components, you would also require the following. You will need to determine what parameters they require and if they need to return back a value.

* **ReadIntoArray(…)**

This method should read the data from the MovieRevenue.txt file into two 1-D array (one for movie title and the other for movie sale figures).

(**Hint:** use character delimiter and **Split**() method)

* **SelectionSortByName(…)**

This method will sort **any sized** 1-D string array using a selection sort in ***ascending*** order. ***Note, DO NOT*** pass the size of the array as a parameter to the method ***OR*** use any global variable/constant that has the size of the array. **Your method MUST be able to determine the size of the array**.

* **SelectionSortByRevenue(…)**

This method will sort **any sized** 1-D double array using a selection sort in ***descending*** order. ***Note, DO NOT*** pass the size of the array as a parameter to the method ***OR*** use any global variable/constant that has the size of the array. **Your method MUST be able to determine the size of the array**.

* **BinSrch(…)**

This method will **perform a binary search** on **any sized** 1-D double array. Please **use this method to determine if there are any movie title with that exact revenue amount**. ***Note, DO NOT*** pass the size of the array as a parameter to the method ***OR*** use any global variable/constant that has the size of the array. **Your method MUST be able to determine the size of the array**.

* **DisplayRevenue(…)**

This method will display the data into the listbox.

**Others**

You will need to perform **linear search** to see if the search string is within any of the string titles. This can be done as part of the search button rather than as a method.

(**Hint** use the string **Contains**() method)

You can use a **global constant for the size of the array during the creation of the array**, **BUT DO NOT use it for the above methods**.

You are free to create any additional methods you may require.

**What to hand in:**

1. Front cover page with your name (your partner’s name), course, assignment number, instructor’s name, section and due date. (If you did this assignment with a partner, then hand in **ONE** piece of work with **BOTH** your names on it).
2. Hardcopy of the program listing of **Form1 cs**
3. Either zip up all files, call it **CSIS1175\_YourSecNumberW2018yourNameAS4.zip and upload via File submission through Blackboard**

**OR**

Hand in Disk containing all necessary **source** files.

**(If you did this with a partner, then please hand in one piece of work with both your names on the cover page.)**

**PLEASE ENSURE THAT THIS IS YOUR OWN WORK OTHERWISE MARKS WILL BE LOST AND WORSE !!!!!!**