**Your favorite 5 movies**

Start a new console solution. Underneath the Program class, add a Movie class with 3 properties, provide a constructor

In Program’s Main method, code a loop that runs 5 times, asking the user to enter

* Movie title
* Movie Genre
* Rating (1 bad to 5 great)

For each movie, create a Movie object and store the objects in an array.

Then create a new **static** class Sort

This class should have 2 **static** methods.

ByGenre(Movie[] inputArray)

ByRating(Movie[] inputArray)

Each of those methods should sort the array passed into them. I am not specifying how you achieve the sort of the Movie objects, you need to figure that out. See my note below.

Back in your Main method, after you have created the array of 5 Movie objects,

Call the ByGenre method, and then write out to the console the list of 5 movies, including the title, genre, and rating, sorted by genre.

Next, write a few blank lines, and then …

Call the ByRating method, and then write out to the console the list of 5 movies, including the title, genre, and rating, sorted by rating.

=======================================================================

If you look up how to sort and array of objects by one of the object’s properties, all the “easy” ways require C# constucts that we have not yet learned (ICompare Interface or LINQ). But the simplest sort (also, the most inefficient sort) can be a path I suggest you pursue. See this article:

<https://www.c-sharpcorner.com/UploadFile/3d39b4/bubble-sort-in-C-Sharp/>

That example sorts a simple integer array, not an array of objects, but you should understand how it works. It uses a flag variable as an optimization. The code uses that flag to signal if it did a pass through the array and no changes were required, so it can stop looping. You can remove that flag and it works fine. Here is my version of a bubble sort based on a string property of objects in an array

for (int i = 1; (i <= (pMovieArray.Length - 1)); i++)

{

for (int j = 0; j < (pMovieArray.Length - 1); j++)

{

if ( pMovieArray[j + 1].Genre.CompareTo(pMovieArray[j].Genre) < 0 )

{

Movie temp = pMovieArray[j];

pMovieArray[j] = pMovieArray[j + 1];

pMovieArray[j + 1] = temp;

}

}

}

You should be able to use that to get the Genre sort working. Then for the rating sort, you just have to change this line

if ( pMovieArray[j + 1].Genre.CompareTo(pMovieArray[j].Genre) < 0 )

to make it work for the Rating sort, which is based on integers rather than strings.

=================================================================

*Here is a sample run of my version*

Please enter you 5 favorite movies.

Enter movie title: Moonstuck

Enter the genre: comedy

Please enter a rating from 1 bad to 5 good: 5

Enter movie title: Titanic

Enter the genre: drama

Please enter a rating from 1 bad to 5 good: 3

Enter movie title: Dune

Enter the genre: scifi

Please enter a rating from 1 bad to 5 good: 4

Enter movie title: Raising Arizona

Enter the genre: comedy

Please enter a rating from 1 bad to 5 good: 5

Enter movie title: Ender's Game

Enter the genre: scifi

Please enter a rating from 1 bad to 5 good: 4

Here are the movies sorted by Genre

Title: Moonstuck, Genre: comedy, Rating: 5

Title: Raising Arizona, Genre: comedy, Rating: 5

Title: Titanic, Genre: drama, Rating: 3

Title: Dune, Genre: scifi, Rating: 4

Title: Ender's Game, Genre: scifi, Rating: 4

Here are the movies sorted by Rating

Title: Moonstuck, Genre: comedy, Rating: 5

Title: Raising Arizona, Genre: comedy, Rating: 5

Title: Dune, Genre: scifi, Rating: 4

Title: Ender's Game, Genre: scifi, Rating: 4

Title: Titanic, Genre: drama, Rating: 3