Problem Description

Song Group - Requirement 1

Your dad uses an old computer. He is unaware of the latest smartphones and apps present in it. He used to store his songs in a folder structure organized based on the type of the song. You being an aspirant programmer wanted to create an application which will be useful to download songs and maintain a playlist. There are two major domains Song and Playlist. The song details are stored in Song domain and are grouped together in Playlist domain.

Requirement 1:

Let's start off by creating two Song objects and check whether they are equal.

1. Create a **Song** Class with the following attributes:

Member Field Name	**Type**
_name	string
_artist	string
_songType	string
_rating	double
_numberOfDownloads	int
_dateDownloaded	DateTime

2. Mark all the attributes as private
3. Include appropriate properties.
4. Add a default constructor and a parameterized constructor to take in all attributes in the given order:
   Song(string _name, string _artist, string _songType, double _rating,int _numberOfDownloads,DateTime _dateDownloaded)
5. When the "Song" object is printed, it should display the following details: **[Override the ToString method]**
   Print format:
   Name:"name"
   Artist:"artist"
   Song Type:"songType"
   Rating:"rating"
   Number of Downloads:"numberOfDownloads"
   Date Downloaded:"dateDownloaded"(dd-MM-yyyy)
6. Two songs are considered same if they have the same name, artist, and songType. Implement the logic in the appropriate function. (Case – Insensitive) **[Override the Equals method]**

The input format consists of song details separated by comma in the below order, (**_name,_artist, _songType,_rating,_numberOfDownloads,_dateDownloaded**)

The Input to your program would be details of two songs, you need to display their details as given in "5th point(refer above)" and compare the two songs and display if the Songs are same or different.

Problem Description:
Display the rating as one decimal point.

Note: There is an empty line between display statements. Print the empty lines in the main method.

Sample INPUT & OUTPUT 1:

Enter song 1 detail:
Shape of you,Edsheeran,Pop,4.8,100000,12-01-2018
Enter song 2 detail:
Shape of you,Edsheeran,Pop,4.8,100000,12-01-2018

Song 1:
Name:Shape of you
Artist:Edsheeran
Song Type:Pop
Rating:4.8
Number of Downloads100000
Date Downloaded:12-01-2018

Song 2:
Name:Shape of you
Artist:Edsheeran
Song Type:Pop
Rating:4.8
Number of Downloads100000
Date Downloaded:12-01-2018

Song 1 is same as Song 2

Sample INPUT & OUTPUT 2:

Enter song 1 detail:
Shape of you,Edsheeran,Pop,4.8,100000,12-01-2018
Enter song 2 detail:
Perfect,Edsheeran,Pop,4.8,100000,12-01-2018

Song 1:
Name:Shape of you
Artist:Edsheeran
Song Type:Pop
Rating:4.8
Number of Downloads100000
Date Downloaded:12-01-2018

Song 2:
Name:Perfect
Artist:Edsheeran
Song Type:Pop
Rating:4.8
Number of Downloads100000
Date Downloaded:12-01-2018

Song 1 and Song 2 are different

Song Group - Requirement 2

Requirement 2:
Now we are going to start creating a playlist and add songs to it. Start with creating a playlist and use menu-driven approach to add, remove details of the songs in the group.

a)Create a Class **Song** with the following attributes:

Member Field Name	**Type**
_name	string
_artist	string
_songType	string
_rating	double
_numberOfDownloads	int
_dateDownloaded	DateTime

Mark all the attributes as private.
Include appropriate properties.
Add a default constructor and a parameterized constructor to take in all attributes in the given order: **Song(string _name, string _artist, string _songType, double _rating, int _numberOfDownloads,DateTime _dateDownloaded)**

b)Create a Class **PlayList** with the following attributes:

Member Field Name	**Type**
_name	string
_songList	List<Song>

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: **PlayList(string _name, List<Song> _songList).** In constructor pass the songList value as an empty list. Only one playlist will be present at a time.

c) Create the following static method in Song class,

Method Name	Description
public static Song CreateSong(string song)	This method accepts a string which contains song details separated by commas. Split the details and create a song object from the details and return it.

The song details should be given as a comma-separated value in the below order,
name,artist,songType, rating, numberOfDownloads,dateDownloaded

d) Create the following methods in PlayList class,

Method Name	Description
public void AddSongToPlaylist(Song song)	This method accepts a Song object and add the song to the song list of the current Playlist.
public bool RemoveSongFromPlaylist(String name)	This method acceptst a string . The name of the song is passed as value. Delete the song with the specified name from the current PlayList. If a song with the given song name found, delete the song and return **true**. If a songwith the song name is not found return **false**.
public void DisplaySongs()	This method will display the song list in the current playlist. If the song list is empty display "**No song to show"**, else display "Songs in [playlist name]" and display all the song details in the specified format. Where [playlist name] specifies the name of the playlist.

After deletion, if true is returned print "**Song successfully deleted**", else print "**Song not found in the Play List**".

Note: The above print statements should be present in the main method. The songList of a Playlist is printed in the same order as they added to the playlist.

When the “song” object is printed, it should display the following format
Print format:
Console.WriteLine("{0} {1,15} {2,15} {3,15} {4,15} {5,15}", "Name","Artist","Song Type","Rating","No of Download","Date Downloaded");

Sample Input and Output:

Enter the Play list name:
Mysongs
1.Add Song
2.Remove Song

3.Display
4.Exit
Enter your choice:
3
No song to show
1.Add Song
2.Remove Song
3.Display
4.Exit
Enter your choice:
1
Enter the number of Songs:
3
Enter song 1 detail:
Shape of you , Edsheeran ,Pop ,4.8,12000,12-04-2017
Enter song 2 detail:
Perfect, Edsheeran ,Pop,4.5,120500,06-08-2017
Enter song 3 detail:
Mercy ,Shawn Mendes ,Jazz ,4.6,50000,03-09-2016
1.Add Song
2.Remove Song
3.Display
4.Exit
Enter your choice:
3
Songs in Mysongs

Name	Artist	Song Type	Rating	No of Download	Date Downloaded
Shape of you	Edsheeran	Pop	4.8	12000	12-04-2017
Perfect	Edsheeran	Pop	4.5	120500	06-08-2017
Mercy	Shawn Mendes	Jazz	4.6	50000	03-09-2016

1.Add Song
2.Remove Song
3.Display
4.Exit
Enter your choice:
2
Enter the name of the song to be deleted:
Perfect
Song successfully deleted
1.Add Song
2.Remove Song
3.Display
4.Exit
Enter your choice:
3
Songs in Mysongs

Name	Artist	Song Type	Rating	No of Download	Date Downloaded
Shape of you	Edsheeran	Pop	4.8	12000	12-04-2017
Mercy	Shawn Mendes	Jazz	4.6	50000	03-09-2016

1.Add Song
2.Remove Song
3.Display
4.Exit
Enter your choice:
4

Song Group - Requirement 3

Requirement 3:

In this requirement develop a feature in which you can search a List of Songs by song type, date of download or rating.

a) Create a Class **Song** with the following attributes:

Member Field Name	**Type**
_name	string
_artist	string
_songType	string
_rating	double
_numberOfDownloads	int
_dateDownloaded	DateTime

Mark all the attributes as private.
Include appropriate properties.
Add a default constructor and a parameterized constructor to take in all attributes in the given order: Song(string _name, string _artist, string _songType, double _rating,int _numberOfDownloads,DateTime _dateDownloaded)

b) Create a class **SongBO** with the following methods,

Method Name	**Description**
public List<Song> FindSong(List<Song> songList,string type)	This method accepts a list of songs and songType as arguments and returns a list of songs with the given songType [Case-sensitive].
public List<Song> FindSong(List<Song>SongList,DateTime dateCreated)	This method accepts a list of songs and created date as arguments and returns a list of songs that were created on the given specified date.
public List<Song>FindSong(List<Song>SongList,double rating)	This method accepts a list of songs and rating as arguments. Find all the songs with the given rating from the song list and return it.

The song details should be given as a comma-separated value in the below order, name,artist,songType,rating, numberOfDownloads,dateDownloaded

When the “song” object is printed, it should display the following details
Print format:
Console.WriteLine("{0} {1,15} {2,15} {3,15} {4,15} {5,15}", "Name", "Artist", "Song Type", "Rating", "No of Download", "Date of Download");

Note: The song lists are displayed in the main method.
If any other choice is selected, display "**Invalid choice**"

Sample Input and Output 1:

Enter the number of Songs:
4
Shape Of You,Edsheeran,Pop,4.8,10000,12-10-2017
Perfect,Edsheeran,Pop,4.5,12000,04-05-2016
Something just like this,Coldplay,Melody,4.5,105421,05-07-1997
Mercy,Shawn Mendes,Jazz,4.8,214500,04-05-2016
Enter a search type:
1.Song Type
2.Date of Download
3.Rating
1
Enter the type:
Pop

Name	Artist	Song Type	Rating	No of Download	Date Downloaded
Shape Of You	Edsheeran	Pop	4.8	10000	12-10-2017
Perfect	Edsheeran	Pop	4.5	12000	04-05-2016

Sample Input and Output 2:

Enter the number of Songs:
4
Shape Of You,Edsheeran,Pop,4.8,10000,12-10-2017
Perfect,Edsheeran,Pop,4.5,12000,04-05-2016
Something just like this,Coldplay,Melody,4.5,105421,05-07-1997
Mercy,Shawn Mendes,Jazz,4.8,214500,04-05-2016
Enter a search type:
1.Song Type
2.Date of Download
3.Rating
2
Enter the date:
04-05-2016

Name	Artist	Song Type	Rating	No of Download	Date Downloaded
Perfect	Edsheeran	Pop	4.5	12000	04-05-2016
Mercy	Shawn Mendes	Jazz	4.8	214500	04-05-2016

Sample Input and Output 3:

Enter the number of Songs:
4
Shape Of You,Edsheeran,Pop,4.8,10000,12-10-2017
Perfect,Edsheeran,Pop,4.5,12000,04-05-2016
Something just like this,Coldplay,Melody,4.5,105421,05-07-1997
Mercy,Shawn Mendes,Jazz,4.8,214500,04-05-2016
Enter a search type:
1.Song Type
2.Date of Download
3.Rating
3
Enter the rating:
4.5

Name	Artist	Song Type	Rating	No of Download	Date Downloaded
Perfect	Edsheeran	Pop	4.5	12000	04-05-2016
Something just like this	Coldplay	Melody	4.5	105421	05-07-1997

Sample Input and Output 4:

Enter the number of contact details:
4
Shape Of You,Edsheeran,Pop,4.8,10000,12-10-2017
Perfect,Edsheeran,Pop,4.5,12000,04-05-2016
Something just like this,Coldplay,Melody,4.5,105421,05-07-1997
Mercy,Shawn Mendes,Jazz,4.8,214500,04-05-2016
Enter a search type:
1.Name
2.Date created
3.Email domain
4
Invalid choice

Playlist - Requirement 4

Requirement 4:

In this requirement, you need to sort the list of song based on name, rating or popularity.

a) Create a Class **Song** with the following attributes:

Member Field Name	**Type**
_name	string
_artist	string
_songType	string
_dateDownloaded	DateTime
_rating	double
_numberOfDownloads	int

Mark all the attributes as private.
Include appropriate properties.
Add a default constructor and a parameterized constructor to take in all attributes in the given order: Song(string _name, string _artist, string _songType,DateTime _dateDownloaded, double _rating, int _numberOfDownloads)

b) Create the following static methods in the Song class,

Method Name	**Description**
static Song CreateSong(string detail)	This method accepts a String. The song detail separated by commas is passed as the argument. This method will split the details and creates a song object and returns it.

The input format of Song details is separated by comma in the below order,
(_name, _artist, _songType, _dateDownloaded, _rating, _numberOfDownloads)

c) The Song class should implement the IComparable interface which sorts the Song list based on names. While comparing all the names in the list are unique.

d) Create a class **RatingComparer** which implements IComparer interface and sort the song list based on rating.

e) Create a class **PopularityComparer** which implements IComparer interface and sort the song list based on the number of downloads.

Get the number of songs and song details and create a song list. Sort the song according to the given option and display the list.

When the “song” object is printed, it should display the following details

Print format:
Console.WriteLine("{0} {1,15} {2,15} {3,15} {4,15} {5,15}", "Name","Artist","Song Type","Date of Download","Rating","No of Downloads");

Sample Input and Output 1:

Enter the number of the songs:
4
The Way,Fastball,Emotional,12-09-2017,4,10000
Mad World,Gary Jules,Emotional,12-09-2017,4.2,15000
Let It Be,The Beatles,Motivational,11-01-2010,4.8,500000
Amazed,lone star,Country,12-09-2000,4.2,10000
Enter a type to sort:
1.Sort by name
2.Sort by Rating
3.Sort by Popularity
1

Name	Artist	Song Type	Date of Download	Rating	No of Downloads
Amazed	lone star	Country	12-09-2000	4.2	10000
Let It Be	The Beatles	Motivational	11-01-2010	4.8	500000
Mad World	Gary Jules	Emotional	12-09-2017	4.2	15000
The Way	Fastball	Emotional	12-09-2017	4.0	10000

Sample Input and Output 2:

Enter the number of the songs:
4
The Way,Fastball,Emotional,12-09-2017,4.5,10000
Mad World,Gary Jules,Emotional,12-09-2017,4.2,15000
Let It Be,The Beatles,Motivational,11-01-2010,4.8,500000
Amazed,lone star,Country,12-09-2000,3.8,10000
Enter a type to sort:
1.Sort by name
2.Sort by Rating
3.Sort by Popularity
2

Name	Artist	Song Type	Date of Download	Rating	No of Downloads
Amazed	lone star	Country	12-09-2000	3.8	10000

Mad World	Gary Jules	Emotional	12-09-2017	4.2	15000
The Way	Fastball	Emotional	12-09-2017	4.5	10000
Let It Be	The Beatles	Motivational	11-01-2010	4.8	500000

Sample Input and Output 3:

Enter the number of the songs:
4
The Way,Fastball,Emotional,12-09-2017,4.5,15000
Mad World,Gary Jules,Emotional,12-09-2017,4.2,10500
Let It Be,The Beatles,Motivational,11-01-2010,4.8,5000
Amazed,lone star,Country,12-09-2000,3.8,10000
Enter a type to sort:
1.Sort by name
2.Sort by Rating
3.Sort by Popularity
3

Name	Artist	Song Type	Date of Download	Rating	No of Downloads
Let It Be	The Beatles	Motivational	11-01-2010	4.8	5000
Amazed	lone star	Country	12-09-2000	3.8	10000
Mad World	Gary Jules	Emotional	12-09-2017	4.2	10500
The Way	Fastball	Emotional	12-09-2017	4.5	15000

Playlist - Requirement 5

Requirement 5:

In this requirement, given a list of songs you need to find the number of songs present in each domain using SortedDictionary.

a) Create a Class **Song** with the following attributes:

Member Field Name	**Type**
_name	string
_artist	string
_songType	string
_dateDownloaded	DateTime
_rating	double
_numberOfDownloads	int

Mark all the attributes as private.
Include appropriate properties.
Add a default constructor and a parameterized constructor to take in all attributes in the given order: Song(String name, String artist, String songType, Date dateDownloaded, Double rating ,Integer numberOfDownloads).

b) Create the following static methods in the Song class,

Method Name	**Description**
static SortedDictionary<string,int> CalculateTypeCount(List<Song> list)	This method accepts a list of Songs as arguments and returns a SortedDictionary with the songType as key and number of songs for the particular songType as value and return the sortedDictionary.

In the SortedDictionary have the songType as key and Count the number of songs for the type and keep the number of songs for the type as value. Print the value sorted by the songType.

The song details should be given as a comma separated value in the below order,
_name, _artist, _songType, _dateDownloaded, _rating, _numberOfDownloads

Print format:
Console.WriteLine("{0} {1,15}","Song type","Count");

Sample Input and Output 1:

Enter the number of songs
4
Shape of you,Ed sheeran,Pop,20-01-2018,4.5,150000
Sorry,Justin Bieber,Pop,15-12-2017,4.7,10000
We will rock you,Queen,Rock,11-01-2010,4.8,500000
Losing sleep,Chris,Country,12-09-2017,4,10000

Song type	Count
Country	1
Pop	2
Rock	1

Playlist - Requirement 6

Requirement 6:

In this requirement, predict the state of mind of the user. The state of mind depends on the song type the user listened recently. Given the list of songs, he heard recently predict the mind state of the user.

A user is happy if he has heard many songs of type Celebration.
A user is depressed if he has heard many songs of type Emotional.
A user is energetic if he has heard many songs of type Motivational.

a) Create a **Song** Class with the following private attributes:

Member Field Name	**Type**
_name	string
_artist	string
_songType	string
_dateDownloaded	DateTime
_rating	double
_numberOfDownloads	int

Mark all the attributes as private.
Include appropriate properties.
Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: **Song(string _name, string _artist, string _songType,DateTime _dateDownloaded, double _rating, int _numberOfDownloads)**

b) Create the following static methods in **Song** class,

Method Name	**Description**
static Dictionary<string,int> CalculateTypeCount(List<Song> list)	This method accepts a list of Songs as arguments and create a Dictionary with the songType as key and number of songs for the particular songType as value and return the Dictionary.
static string PredictState(Dictionary<string,int> perTypeCount)	This method accepts the song type Dictionary as argument and returns the state of mind as given below, if song type is Emotional -> return "depressed"

	if song type is Celebration -> return "happy" if song type is Motivational -> return "energetic"

The input format of Song details is separated by comma in the below order,
(_name, _artist, _songType, _dateDownloaded, _rating, _numberOfDownloads)

Note: The statement "**The user is feeling [state of mind]"** is displayed in the main method. Where **[state of mind]**represents the mind state of the user, which is returned by predictState method.

Sample INPUT & OUTPUT 1:

Enter the number of songs
4
Here comes the sun,The Beatles,Motivational,20-01-2018,4.5,150000
Don't Give Up,Peter Gabriel,Motivational,15-12-2017,4.7,10000
Let It Be,The Beatles,Motivational,11-01-2010,4.8,500000
Mad World,Gary Jules,Emotional,12-09-2017,4,10000
The user is feeling energetic

Sample INPUT & OUTPUT 2:

Enter the number of songs
5
Mad World,Gary Jules,Emotional,12-09-2017,4.2,15000
Someone Like You,Adele,Emotional,15-12-2017,4.7,10000
Hurt,Johnny Cash,Emotional,11-01-2010,4.8,500000
The Way,Fastball,Emotional,12-09-2017,4,10000
Here comes the sun,The Beatles,Motivational,20-01-2018,4.5,150000
The user is feeling depressed