**Mail Folder - Requirement 1**

You work for a start-up company. The company has been using an outdated mail system. Looking at the difficulty in the old system, the management decided to build their own mail server. There are two major domains, Mail and Mail Folder. The mail details are stored in the mail domain. The mail folder is used to group the emails together.  
  
**Requirement 1:**

Let’s start off by creating two Mail objects and check whether they are equal.

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

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| \_id | long |
| \_to | string |
| \_from | string |
| \_subject | string |
| \_content | string |
| \_receivedDate | DateTime |
| \_size | double |

1. Mark all the attributes as private
2. Include appropriate properties.
3. Add a default constructor and a parameterized constructor to take in all attributes in the given order: Mail(long \_id, string \_to, string \_from, string \_subject, string \_content, DateTime \_receivedDate,double \_size)
4. When the “mail” object is printed, it should display the following details: **[Override the ToString method]**  
   Print format:  
   Id:"\_id"  
   To:"\_to"  
   From:"\_from"  
   Subject:"\_subject"  
   Content:"\_content"  
   Received Date:"\_receivedDate"  
   Size:"\_size"
5. Two emails are considered same if they have the same to address, from address, and subject. Implement the logic in the appropriate function. (Case – Insensitive) **[Override the Equals method]**  
     
   The input format consists of mail details separated by comma in the below order,  
   (\_**id,\_to, \_from,\_subject,\_content,\_receivedDate,\_size**)

The Input to your program would be details of two emails, you need to display their details as given in "5th point(refer above)" and compare the two emails and display if the Mails are same or different.  
  
**Note:**There is an empty line between display statements. Print the empty lines in Main function.  
              Display one digit after the decimal point for double datatype.  
  
**Sample INPUT & OUTPUT 1:**  
  
Enter mail 1 detail:  
**1001,meyyappan@gmail.com,satish@gmail.com,Master Copy,Attached doc,05-05-2017,10.0**  
Enter mail 2 detail:  
**1001,meyyappan@gmail.com,satish@gmail.com,Master Copy,Attached doc,05-05-2017,10.0**  
  
Mail 1:  
Id:1001  
To:meyyappan@gmail.com  
From:satish@gmail.com  
Subject:Master Copy  
Content:Attached doc  
ReceivedDate:05-05-2017  
Size:10.0  
  
Mail 2:  
Id:1001  
To:meyyappan@gmail.com  
From:satish@gmail.com  
Subject:Master Copy  
Content:Attached doc  
ReceivedDate:05-05-2017  
Size:10.0  
  
Mail 1 is same as Mail 2  
  
  
**Sample INPUT & OUTPUT 2:**  
  
Enter mail 1 detail:  
**1001,meyyappan@gmail.com,satish@gmail.com,Master Copy,Attached doc,05-05-2017,10.0**  
Enter mail 2 detail:  
**1002,satish@gmail.com,meyyappan@gmail.com,Master Copy,Attached doc,05-05-2017,10.0**  
  
Mail 1:  
Id:1001  
To:meyyappan@gmail.com  
From:satish@gmail.com  
Subject:Master Copy  
Content:Attached doc  
ReceivedDate:05-05-2017  
Size:10.0  
  
Mail 2:  
Id:1002  
To:satish@gmail.com  
From:meyyappan@gmail.com  
Subject:Master Copy  
Content:Attached doc  
ReceivedDate:05-05-2017  
Size:10.0  
  
Mail 1 and Mail 2 are different

**Mail Folder - Requirement 2**

**Requirement 2:**  
Now we are gonna start creating a folder and add mail to it. Start with creating a folder and use menu-driven approach to add, remove, display details of the mail in the folder.  
  
a)Create a Class **Mail** with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| \_id | long |
| \_from | string |
| \_to | string |
| \_subject | string |
| \_content | string |
| \_receivedDate | DateTime |
| \_size | double |

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:  **public Mail(long \_id, string \_from, string \_to, string \_subject, string \_content, DateTime \_receivedDate, double \_size)**  
  
b)Create a Class **MailFolder** with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| \_name | string |
| \_mailList | List<Mail> |

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:**MailFolder(string \_name, List<Mail> \_mailList).**In constructor pass the \_mailList value as an empty list. Only one folder will be present at a time.  
  
c) Create the following static method in **Mail**class,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public static Mail CreateMail(string detail) | This method accepts a string which contains mail details separated by commas. Split the details and create a mail object from the details and return it. |

The mail details should be given as a comma-separated value in the below order,  
**\_id,\_from,\_to, \_subject, \_content, \_receivedDate, \_size**  
  
d) Create the following methods in **MailFolder**class,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public void AddMailToFolder(Mail mail) | This method accepts a mail object and add the mail to the mail list of the current mail folder. |
| public boolRemoveMailFromFolder(long id) | This method will get the id of the mail and delete the mail with the specified \_id from the current folder. If a mail with the given \_id found, delete the mail and return **true**. If a mail with the \_id is not found return **false**. |
| public void DisplayMails() | This method will display the mail list in the current folder. If the mail list is empty display "**No mails to show"**, else display "Mails in [folder name]" and display all the mail details in the specified format. Where [folder name] specifies the name of the folder. |

After deletion, if true is returned print "**Mail successfully deleted**", else print "**Mail not found in the folder**". After adding mail to the folder, print "**Mail successfully added**".  
  
**Note:** The above print statements should be present in the Main method.  
Display double values correct to one decimal place.  
  
When the “mail” object is printed, it should display the following format  
Print format:  
**Console.WriteLine("{0} {1,15} {2,15} {3,15} {4,15} {5,15} {6,15}", "Id", "From", "To", "Subject", "Content", "Received Date", "Size");**  
**Display 1 digit after decimal point in Double.**  
  
**Sample Input and Output:**  
  
Enter the name of the folder:  
**Inbox**  
1.Add Mail  
2.Delete Mail  
3.Display Mails  
4.Exit  
Enter your choice:  
**3**  
No mails to show  
1.Add Mail  
2.Delete Mail  
3.Display Mails  
4.Exit  
Enter your choice:  
**1**  
Enter the details of mail in CSV format:  
**12,john@abc.in,jane@abc.in,Hi,Happy New Year,01-01-2018,10**  
Mail successfully added  
1.Add Mail  
2.Delete Mail  
3.Display Mails  
4.Exit  
Enter your choice:  
**1**  
Enter the details of mail in CSV format:  
**16,jack@abc.in,jane@abc.in,Hi,Happy Pongal,14-01-2018,15**  
Mail successfully added  
1.Add Mail  
2.Delete Mail  
3.Display Mails  
4.Exit  
Enter your choice:  
**3**  
Mails in Inbox  
Id        From           To             Subject        Content             Received Date  Size        
12        john@abc.in    jane@abc.in    Hi             Happy New Year      01-01-2018     10.0        
16        jack@abc.in    jane@abc.in    Hi             Happy Pongal        14-01-2018     15.0        
1.Add Mail  
2.Delete Mail  
3.Display Mails  
4.Exit  
Enter your choice:  
**2**  
Enter the id of the mail to be deleted:  
**13**  
Mail not found in the folder  
1.Add Mail  
2.Delete Mail  
3.Display Mails  
4.Exit  
Enter your choice:  
**2**  
Enter the id of the mail to be deleted:  
**16**  
Mail successfully deleted  
1.Add Mail  
2.Delete Mail  
3.Display Mails  
4.Exit  
Enter your choice:  
**3**  
Mails in Inbox  
Id        From           To             Subject        Content             Received Date  Size        
12        john@abc.in    jane@abc.in    Hi             Happy New Year      01-01-2018     10.0        
1.Add Mail  
2.Delete Mail  
3.Display Mails  
4.Exit  
Enter your choice:  
**4**

In this requirement, you need to sort the list of mails based on from address, receivedDate or size.  
  
a) Create a Class **Mail**with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| \_id | long |
| \_from | string |
| \_to | string |
| \_subject | string |
| \_content | string |
| \_receivedDate | DateTime |
| \_size | double |

Mark all the attributes as private.  
Include appropriate getters and setters.  
Add a default constructor and a parameterized constructor to take in all attributes in the given order: Mail(long \_id, string \_from,string \_to, string \_subject, string \_content, DateTime \_receivedDate, double \_size)  
  
b) Create the following static methods in the Mail class,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| static Mail createMail(string detail) | This method accepts a String. The mail detail separated by commas is passed as the argument. Split the details and create a mail object and returns it. |

The mail details should be given as a comma-separated value in the below order,  
\_id,\_from,\_to,\_subject,\_content,\_receivedDate,\_size  
  
c) The Mail class should implement the **Comparable** interface which sorts the Mail list based on from \_address. While comparing, all the from \_address in the list are unique.  
  
d) Create a class **DateComparer** which implements **Comparator**interface and sort the Mail list based on \_receivedDate. While comparing, all the \_receivedDate attributes in the list are unique.  
  
e) Create a class **SizeComparer** which implements **Comparator** interface and sort the Mail list based on the \_size. While comparing, all the \_size attributes in the list are unique.

Get the number of Mail and mail details and create a mail list. Sort the mail according to the given option and display the list.  
  
When the “mail” object is printed, it should display the following details  
Print format:  
**"Id", "From", "To", "Subject", "Content", "Received date", "Size"**  
  
Display one digit after decimal point for Double datatype.  
  
**Sample Input and Output 1:**  
  
Enter the number of mails:  
**4  
1,raj@abc.in,bala@abc.in,Freshers' list,PFA the Freshers db,05-01-2018,256  
2,amir@abc.in,chris@abc.in,M.O.M,PFA the M.O.M,29-01-2018,678  
3,abdul@abc.in,antony@abc.in,project requirement,PFA the requirements,20-01-2018,1658  
4,karim@abc.in,krishna@abc.in,Accounts,PFA the accounts,27-01-2018,2048**  
Enter a type to sort:  
1.Sort by from address  
2.Sort by date received  
3.Sort by size  
**1**  
Id       From                      To                               Subject                              Content                                  Received date     Size  
3        abdul@abc.in    antony@abc.in   project requirement    PFA the requirements    20-01-2018            1658.0  
2        amir@abc.in       chris@abc.in        M.O.M                                 PFA the M.O.M                   29-01-2018            678.0  
4        karim@abc.in     krishna@abc.in  Accounts                          PFA the accounts             27-01-2018            2048.0  
1        raj@abc.in            bala@abc.in       Freshers' list                    PFA the Freshersdb       05-01-2018            256.0  
  
**Sample Input and Output 2:**  
  
Enter the number of mails:  
**4  
1,raj@abc.in,bala@abc.in,Freshers' list,PFA the Freshers db,05-01-2018,256  
2,amir@abc.in,chris@abc.in,M.O.M,PFA the M.O.M,29-01-2018,678  
3,abdul@abc.in,antony@abc.in,project requirement,PFA the requirements,20-01-2018,1658  
4,karim@abc.in,krishna@abc.in,Accounts,PFA the accounts,27-01-2018,2048**  
Enter a type to sort:  
1.Sort by from address  
2.Sort by date received  
3.Sort by size  
**2**  
Id       From                      To                               Subject                              Content                                  Received date     Size  
1        raj@abc.in            bala@abc.in       Freshers' list                    PFA the Freshersdb       05-01-2018            256.0  
3        abdul@abc.in    antony@abc.in   project requirement    PFA the requirements    20-01-2018            1658.0  
4        karim@abc.in     krishna@abc.in  Accounts                          PFA the accounts             27-01-2018            2048.0  
2        amir@abc.in       chris@abc.in        M.O.M                                 PFA the M.O.M                   29-01-2018            678.0  
  
**Sample Input and Output 3:**  
  
Enter the number of mails:  
**4  
1,raj@abc.in,bala@abc.in,Freshers' list,PFA the Freshers db,05-01-2018,256  
2,amir@abc.in,chris@abc.in,M.O.M,PFA the M.O.M,29-01-2018,678  
3,abdul@abc.in,antony@abc.in,project requirement,PFA the requirements,20-01-2018,1658  
4,karim@abc.in,krishna@abc.in,Accounts,PFA the accounts,27-01-2018,248**  
Enter a type to sort:  
1.Sort by from address  
2.Sort by date received  
3.Sort by size  
**3**  
Id       From                      To                               Subject                              Content                                  Received date     Size  
4        karim@abc.in     krishna@abc.in  Accounts                          PFA the accounts             27-01-2018            248.0  
1        raj@abc.in            bala@abc.in       Freshers' list                    PFA the Freshersdb       05-01-2018            256.0  
2        amir@abc.in       chris@abc.in        M.O.M                                 PFA the M.O.M                   29-01-2018            678.0  
3        abdul@abc.in    antony@abc.in   project requirement    PFA the requirements    20-01-2018            1658.0