**Hotel - Requirement 1**

Create a simple application in which User can store the details of the hotel and rooms available in it. There are two major domains Hotel and Room.

**Requirement 1:**

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

1. Create a class **Room** with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| \_number | int |
| \_floor | int |
| \_type | string |
| \_capacity | int |
| \_bookedTime | DateTime |
| \_price | double |

1. Mark all the attributes as private.
2. Create / Generate appropriate properties.
3. Add a default constructor and a parameterized constructor to take in all attributes in the given order:

**Room(int \_number, int \_floor, string \_type, int \_capacity,DateTime \_bookedTime, double \_price)**

1. When the “room” object is printed, it should display the following details: **[Override the ToString**

**method]**

**Print format:**

Number: "\_number"

Floor: "\_floor"

Type: "\_type"

Capacity: "\_capacity"

Booked Time: "\_bookedTime" Price: "\_price"

1. Two rooms are considered same if they have the same floor, number, and type. Implement the logic in the appropriate function. (Case – Insensitive)  **[Override the Equals method]**

Create **Program** class with **Main** method. Get the room details,it is used to access the above class and its method.

All the input and output operations are done in this method.

The input format consists of room details separated by comma in the below order,

**\_number,\_floor,\_type,\_capacity,\_bookedTime,\_price**

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

**Problem Overview:**

The first two line of input consist of a string, that corresponds to the room details(which is comma seperated). Refer above input format.

Display the two room details in Main method using **ToString** method(Refer above format). And also check if the two rooms are same or different(Use Equals method to compare the two objects).

Equals method return bool value to Main method(true or false).

If the Equals method returns true, then print "**Room 1 is same as Room 2**". If the method returns false, then print "**Room 1 and Room 2 are different**".

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

Display one decimal place for double datatype.

**Sample INPUT & OUTPUT 1:**

Enter room 1 detail:

**112,1,Villa,5,27-01-2018 15:00:00,15000** Enter room 2 detail:

**112,1,Villa,5,27-01-2018 15:00:00,15000**

Room 1:

Number: 112

Floor: 1

Type: Villa

Capacity: 5

Booked Time: 27-01-2018 15:00:00 Price: 15000.0

Room 2:

Number: 112

Floor: 1

Type: Villa

Capacity: 5

Booked Time: 27-01-2018 15:00:00

Price: 15000.0

Room 1 is same as Room 2

**Sample INPUT & OUTPUT 2:**

Enter room 1 detail:

**112,1,Villa,5,27-01-2018 15:00:00,15000** Enter room 2 detail: **201,2,Quad,4,11-05-2018 13:30:00,7000**

Room 1:

Number: 112

Floor: 1

Type: Villa

Capacity: 5

Booked Time: 27-01-2018 15:00:00 Price: 15000.0

Room 2:

Number: 201

Floor: 2

Type: Quad

Capacity: 4

Booked Time: 11-05-2018 13:30:00 Price: 7000.0

Room 1 and Room 2 are different

**Hotel - Requirement 2**

**Requirement 2:**

Now we are gonna start creating a hotel and add rooms to it. Start with creating a hotel and use menu-driven approach to add, remove, display details of the room in the hotel.

a)Create a class **Room** with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| \_number | int |
| \_floor | int |
| \_type | string |
| \_capacity | int |
| \_bookedTime | DateTime |
| \_price | 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:

**Room(int \_number,int \_floor, string \_type, int \_capacity, Datetime \_bookedTime,double \_price)**

Override **ToString** method for print the Room details.

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

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| \_name | string |
| \_roomList | List<Room> |

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:

**Hotel(string \_name, List<Room> \_roomList)**

In constructor pass the **roomList** value as an empty list. Only one **Hotel** object will be present at a time.

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public void AddRoomToHotel(Room room) | This method accepts a Room object and adds the Room to the the current hotel. |
| public bool RemoveRoom(int number) | This method will get the room number and delete the room with number from the current hotel.  If a room with the given number found, delete the room and re a room with the number is not found return **false**. |
| public void DisplayRooms() | This method will display the room list in the current hotel. If the room list is empty display "**No rooms to show"**, else display "**Rooms in [hotel name]**" and display all the room deta specified format. Where [hotel name] specifies the name of the |

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public static Room CreateRoom(string detail) | This method accepts a string which contains room details sepa commas. Split the details and create a room object from the de return it. |

c) Create the following static method in Room class,

The input format consists of room details separated by comma in the below order, **(\_number,\_floor,\_type,\_capacity,\_bookedTime,\_price)** d) Create the following methods in Hotel class,

After deletion, if true is returned print "**Room successfully deleted"**, else print "**Room not found in hotel**". After adding a room to the hotel, print "**Room successfully added**". **Note:** The above print statements should be present in the main method.

**Problem Overview:**

In this requirement contains menu driven,

1.Add Room

2.Delete Room

3.Display Rooms

4.Exit

If the user select option-1, then get the room details from the user(which is seperated by comma). Split that room details and create a room object(Use CreateRoom method) and add that room to the room list(which is in Hotel) Use AddRoomToHotel method to add the room to the hotel.

If the user select option-2, then get the room number from the user, If that room will found in the roomList(Hotel), then remove the room from the hotel and print the message "**Room**

**successfully deleted**". If the room not found in the list, then print the message "**Room not**

**found in hotel**". Use RemoveRoom method, that method returns boolean(either true or false) value. Display the message in Main menu based upon the appropriate return type.

If the user select option-3, then display the room details for the hotel(Use following format to print the details). Use DisplayRooms method. If there is no rooms in the list, then display "**No rooms to show**".

When the “room” object is printed, it should display the following format Print format:

**Console.WriteLine("{0,-10}{1,-10}{2,-10}{3,-10}{4,-15}{5,-10}", "Number", "Floor",**

**"Type", "Capacity", "Booked time", "Price");**

Create a Class **Program**  with the main method to implement the functionalities.

Display one decimal place for double datatype.

**Sample Input and Output:**

Enter the name of the Hotel:

**Hotel Grand**

1.Add Room

2.Delete Room

3.Display Rooms

4.Exit

Enter your choice:

**1**

**112,1,Villa,5,27-01-2018 15:00:00,15000** Room successfully added

1.Add Room

2.Delete Room

3.Display Rooms

4.Exit

Enter your choice:

**3**

Rooms in Hotel Grand

Number Floor Type Capacity Booked time Price

112 1 Villa 5 27-01-18 15:00:00 15000

1.Add Room

2.Delete Room

3.Display Rooms

4.Exit

Enter your choice:

**2**

Enter the room number to be deleted:

**156**

Room not found in hotel

1.Add Room

2.Delete Room

3.Display Rooms

4.Exit

Enter your choice:

**2**

Enter the room number to be deleted:

**112**

Room successfully deleted

1.Add Room

2.Delete Room

3.Display Rooms

4.Exit

Enter your choice: **3**

No rooms to show

1.Add Room

2.Delete Room

3.Display Rooms

4.Exit

Enter your choice: **4**

**Hotel - Requirement 3**

**Requirement 3:**

In this requirement, develop a feature to list the rooms that have to be checked out within the specified stipulated time. The check-out time will be one day after the bookedTime.

a)Create a class **Room** with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| \_number | int |
| \_floor | int |
| \_type | string |
| \_capacity | int |
| \_bookedTime | DateTime |
| \_price | double |

Mark all the attributes as private.

Create / Generate appropriate properties.

Add a default constructor and a parameterized constructor to take in all attributes in the given order: **Room(int \_number,int \_floor, string \_type, int \_capacity, DateTime \_bookedTime, double \_price).**

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public List<Room> CheckoutRooms(List<Room> roomList,DateTime checkoutTime) | This method accepts a list of rooms and checkoutTime as    the given checkoutTime exceeds the bookedTime of the r day or more add them to a list and return it. |

Override **ToString** method for print the Room details.

b) Create the following static method in Room class,

Create **Program** class with **Main** method. Get the number of rooms and details for the corresponding number of rooms.

The input format consists of room details separated by comma in the below order,

**\_number,\_floor,\_type,\_capacity,\_bookedTime,\_price**

If the returned list is empty print "**No rooms to checkout"**, else print the rooms as given in below format

**Problem Overview:**

First get the number of rooms from the user n.

Then get the n number of lines of input(which is comma seperated), Split that string and create a room object and add that room to the room list.

The main theme of this problem is filter the room by the below constraints.

Get the checkout time from the user. Compare the checkout time with the hotel booked time,

If the difference between the user entered checkout time and room's booked time is greater then or equal to 1(24 hours), then that will be add to the filtered list and display the filtered room list.

The above constraint will also consider the time also.

When the “room” object is printed, it should display the following format Print format:

**Console.WriteLine("{0,-7} {1,-7} {2,-20} {3,-10} {4,-20} {5,-10}", "Number","Floor","Type","Capacity","Booked time","Price");**

**Note:** The above print statements should be present in the Main method. Display one decimal place for double datatype.

**Sample Input and Output: 1**

Enter number of rooms

**4**

Enter Rooms details

**329,3,President Suite,1,27-01-2018 06:30:00,25000**

**417,4,Suite,2,27-01-2018 07:00:00,11000**

**101,1,Cabana,6,27-01-2018 18:00:00,20000**

**517,5,Twin,2,27-01-2018 11:15:00,6000**

Enter checkout time

**28-01-2018 23:01:00**

Rooms to be checkedout

Number Floor Type Capacity Booked time Price

329 3 President Suite 1 27-01-18 06:30:00 25000.0

417 4 Suite 2 27-01-18 07:00:00 11000.0

101 1 Cabana 6 27-01-18 18:00:00 20000.0

517 5 Twin 2 27-01-18 11:15:00 6000.0

**Hotel - Requirement 4**

**Requirement 4:**

In this requirement develop a feature in which you can search a List of Rooms by type or price.

1. Create a class **Room** with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| \_number | int |
| \_floor | int |
| \_type | string |
| \_capacity | int |
| \_bookedTime | DateTime |
| \_price | double |

Mark all the attributes as private.

Create / Generate appropriate properties

Add a default constructor and a parameterized constructor to take in all attributes in the given order:

**Room(int \_number,int \_floor,string \_type, int \_capacity,DateTime \_bookedTime, double**

**\_price)**

1. Create a class **RoomBO** with the following methods,

The room details should be given as a comma-separated value in the below order,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public List<Room> FindRoom(List<Room> roomList,string type) | This method accepts a list of rooms and typ arguments and returns a list of rooms that ma given type. |
| public List<Room> FindRoom(List<Room> roomList,double price) | This method accepts a list of rooms and price arguments and returns a list of rooms that ma given price. |

**(\_number,\_floor,\_type,\_capacity,\_bookedTime,\_price)**

When the “room” object is printed, it should display the following details create a class **program** to implement the functionalities.

**Problem Overview:**

First of all get the total number of rooms from the user n.

The next n line of input contains string, that corresponds to the room details(which is comma seperated). Split that string and create a room object and add that room into the roomList(which is maintained in Main method).

The main theme for this requirement is filter the rooms by users choice.

This requirement contains menu driven,

1.By Type

2.By Price

If the user select option-1, then get the type from the user and filter the room list by the hotel type. Use FindRoom(List,string) method.

If the user select option-2, then get the price from the user and filter the room list by the hotel price. Use FindRoom(List,double) method.

Finally display the room details using following format.

Print format:

**Console.WriteLine("{0,-7}{1,-7}{2,-20}{3,-10}{4,-20}{5}", "Number","Floor","Type","Capacity","Booked time","Price");**

**Note:** The room lists are displayed in the main method.

If any other choice is selected, display "**Invalid choice**" If search detail is not found, display "**No such room is present**"

The above statements are displayed in the main method. **Sample Input and Output 1:**

Enter the number of rooms:

**3**

**112,1,Villa,5,27-01-2018 15:00:00,15000**

**201,2,Quad,4,11-05-2018 13:30:00,7000 213,2,Studio,3,12-05-2018 11:45:00,5000** Enter a search type:

1.By Type

2.By Price **1**

Enter the Type:

**Villa**

Number Floor Type Capacity Booked time Price

112 1 Villa 5 27-01-2018 15:00:00 15000

**Sample Input and Output 2:**

Enter the number of rooms:

**3**

**112,1,Villa,5,27-01-2018 15:00:00,15000**

**201,2,Quad,4,11-05-2018 13:30:00,7000 213,2,Studio,3,12-05-2018 11:45:00,5000** Enter a search type:

1.By Type

2.By Price **2**

Enter the Price:

**7000**

Number Floor Type Capacity Booked time Price

201 2 Quad 4 11-05-2018 13:30:00 7000

**Sample Input and Output 3:**

Enter the number of rooms:

**3**

**112,1,Villa,5,27-01-2018 15:00:00,15000**

**201,2,Quad,4,11-05-2018 13:30:00,7000**

**213,2,Studio,3,12-05-2018 11:45:00,5000** Enter a search type:

1.By Type

2.By Price

**20**

Invalid choice

**Hotel - Requirement 5 Requirement 5:**

In this requirement, you need to sort the list of rooms based on capacity, bookedTime or price.

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| static Room CreateRoom(string detail) | This method accepts a String. The room detail separated is passed as the argument. Split the details and create a r and return it. |

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

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| \_number | int |
| \_floor | int |
| \_type | string |
| \_capacity | int |
| \_bookedTime | DateTime |
| \_price | double |

Mark all the attributes as private.

Create / Generate appropriate properties.

Add a default constructor and a parameterized constructor to take in all attributes in the given order: **Room(int \_number, int \_floor, string \_type,int \_capacity, DateTime \_bookedTime, double \_price)**

1. Create the following static methods in the **Room** class,

The room details should be given as a comma-separated value in the below order,

**\_number,\_floor,\_type,\_capacity,\_bookedTime,\_price**

1. The Room class should implement the I**Comparable** interface which sorts the Room list based on capacity attributes. While comparing, all the capacity attributes in the list are unique.
2. Create a class **BookedTimeComparer** which implements **IComparer** interface and sorts the Room list based on bookedTime. While comparing, all the booked time attributes in the list are unique.
3. Create a class **PriceComparer** which implements **IComparer** interface and sorts the Room list based on the price. While comparing, all the price attributes in the list are unique.

Create **Program** class with **Main** method. Get the number of rooms, room details and create a Room list.

Sort the rooms according to the selected sorting option and display the corresponding sorted list.

**Problem Overview:**

First of all get the number of rooms n from the user.

The next n line contains string, that corresponds to the room details(which is comma seperated). Split that string and create a room object and add to the room list(which is maintain in **Main** method). The main theme of this requirement is Sorting the room list by user choice.

This requirement contains menu driven,

1.Sort by capacity

2.Sort by price

3.Sort by booked time

If the user select option-1, then sort the room list based on the capacity of the room(by ascending order).

If the user select option-2, then sort the room list based on the price of the room(by ascending order).

If the user select option-3, then sort the room list based on the booked time of the room(by ascending order).

After sorted the room list display the room list by using the following format.

When the “room” object is printed, it should display the following details. override **ToString** method in Room class.

Print format:

**Console.WriteLine("{0,-7} {1,-7} {2,-20} {3,-10} {4,-20} {5,-5}", "Number","Floor","Type","Capacity","Booked time","Price");**

**Note:** For Double datatype, display one digit after the decimal point.

**Sample Input and Output 1:**

Enter the number of the rooms:

**5**

**112,1,Villa,5,27-01-2018 15:00:00,15000**

**213,2,Studio,3,12-05-2018 11:45:00,5000**

**201,2,Quad,4,11-05-2018 13:30:00,7000**

**329,3,President Suite,1,16-04-2018 12:00:00,25000**

**101,1,Cabana,6,27-07-2018 18:00:00,20000**

Enter a type to sort:

1.Sort by capacity

2.Sort by price

3.Sort by booked time

**1**

Number Floor Type Capacity Booked time Price

329 3 President Suite 1 16-04-2018 12:00:00 25000.0

213 2 Studio 3 12-05-2018 11:45:00 5000.0

201 2 Quad 4 11-05-2018 13:30:00 7000.0

112 1 Villa 5 27-01-2018 15:00:00 15000.0

101 1 Cabana 6 27-07-2018 18:00:00 20000.0

**Sample Input and Output 2:**

Enter the number of the rooms:

**5**

**112,1,Villa,5,27-01-2018 15:00:00,15000**

**213,2,Studio,3,12-05-2018 11:45:00,5000**

**201,2,Quad,4,11-05-2018 13:30:00,7000**

**329,3,President Suite,1,16-04-2018 12:00:00,25000**

**101,1,Cabana,6,27-07-2018 18:00:00,20000**

Enter a type to sort:

1.Sort by capacity

2.Sort by price

3.Sort by booked time

**2**

Number Floor Type Capacity Booked time Price

213 2 Studio 3 12-05-2018 11:45:00 5000.0

201 2 Quad 4 11-05-2018 13:30:00 7000.0

112 1 Villa 5 27-01-2018 15:00:00 15000.0

101 1 Cabana 6 27-07-2018 18:00:00 20000.0

329 3 President Suite 1 16-04-2018 12:00:00 25000.0

**Sample Input and Output 3:**

Enter the number of the rooms:

**5**

**112,1,Villa,5,27-01-2018 15:00:00,15000**

**213,2,Studio,3,12-05-2018 11:45:00,5000**

**201,2,Quad,4,11-05-2018 13:30:00,7000**

**329,3,President Suite,1,16-04-2018 12:00:00,25000**

**101,1,Cabana,6,27-07-2018 18:00:00,20000**

Enter a type to sort:

1.Sort by capacity

2.Sort by price

3.Sort by booked time

**3**

Number Floor Type Capacity Booked time Price

112 1 Villa 5 27-01-2018 15:00:00 15000.0

329 3 President Suite 1 16-04-2018 12:00:00 25000.0

201 2 Quad 4 11-05-2018 13:30:00 7000.0

213 2 Studio 3 12-05-2018 11:45:00 5000.0

101 1 Cabana 6 27-07-2018 18:00:00 20000.0

**Hotel - Requirement 6**

**Requirement 6:**

In this requirement, given a list of Rooms, you need to find the number of rooms booked on a Date using Map.

1. Create a Class Room with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| \_number | int |
| \_floor | int |
| \_type | string |
| \_capacity | int |
| \_bookedTime | DateTime |
| \_price | double |

Mark all the attributes as private.

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| static SortedDictionary<DateTime,int> CalculateDateCount(List<Room> list) | This method accepts a list of Room as arguments and returns a**SortedDictionary** with the \_**bookedDate** as key and numbe booked on the date as value and returns the SortedDiction |

Include appropriate properties.

Add a default constructor and a parameterized constructor to take in all attributes in the given order:

**Room(int \_number, int \_floor, string type,int \_capacity, DateTime \_bookedTime, double**

**\_price)**

1. Create the following static methods in the **Room** class,

In the SortedDictionary have the **\_bookedDate** as key and Count the number of rooms booked on the date and keep the number of Rooms as value. Print the value sorted by **\_bookedDate.**

The Room details should be given as a comma-separated value in the below order, **(\_number,\_floor,\_type,\_capacity,\_bookedTime,\_price)**

**Problem Overview:**

First of all get the number of rooms from the user n.

The next n line of input contains the room details(which is comma seperated). Split that string and create a room object and add to the room list(which is maintain in Main method). The main theme of the requirement is calculate the number of occurance of the room based on the booked time of the room.

Please use SortedDictionary to do this requirement.

Finally display the dictionary details by the following format.

Print format:

**Console.WriteLine("{0,-15}{1}","Date","Count");** create a class **program** with main method to implement the funtionalities.

**Sample Input and Output 1:**

Enter the number of the rooms:

**5**

**112,1,Villa,5,27-01-2018 15:00:00,15000**

**201,2,Quad,4,27-01-2018 13:30:00,7000**

**213,2,Studio,3,28-01-2018 11:45:00,5000**

**329,3,President Suite,1,25-01-2018 12:00:00,25000**

**417,4,Suite,2,24-01-2018 16:00:00,11000**

Date Count

24-01-2018 1

25-01-2018 1

27-01-2018 2

28-01-2018 1