Linq To Objects

**Suitable Course**

John Academy is offering ***n*** number of courses which have a different fee structure. Students find difficulty in choosing the course within their budget. So the academy team decided to get the minimum and maximum fees that the students can afford and display them the courses which lies within that range.  
  
Write a C# program to filter and display the details of the courses between that minimum and maximum (both **inclusive**) fees range using LINQ queries.  
  
**[Note:  Strictly adhere to the object-oriented specifications given as a part of the problem statement.  
Follow the naming conventions as mentioned. Create separate classes in separate files.]**  
  
Consider the class **Course** with following private attributes,

|  |  |
| --- | --- |
| **Data Type** | **Attributes** |
| string | \_id |
| string | \_name |
| string | \_category |
| int | \_price |

The methods for **getters**, **setters** and **constructors** are given in the template code.

Consider the class **Program**. It includes the method Main.  
In the Main method,

* read the course details from the user and add it into arraylist, write a linq query to filter and display the details of the courses between that minimum and maximum fees range.

**Input and Output Format :**  
  
The min and max limit price are **inclusive** for course filter.  
Display the filters course details in **insertion order**.  
If there is no course available in between min and max limit then display “**No course available in this limit**”.  
Refer sample input and output for formatting specifications.   
  
**[All text in bold corresponds to input and rest corresponds to output. ]**  
  
  
**Sample Input And Output 1:**  
Enter the number of Courses  
**5**  
Course 1 Details  
Enter the id  
**1**  
Enter the name  
**Basic java**  
Enter the category  
**Java**  
Enter the price  
**15000**  
Course 2 Details  
Enter the id  
**2**  
Enter the name  
**Advanced html and css**  
Enter the category  
**Web designing**  
Enter the price  
**5000**  
Course 3 Details  
Enter the id  
**3**  
Enter the name  
**Core java**  
Enter the category  
**java**  
Enter the price  
**25000**  
Course 4 Details  
Enter the id  
**4**  
Enter the name  
**C and C++**  
Enter the category  
**Basic programs**  
Enter the price  
**13000**  
Course 5 Details  
Enter the id  
**5**  
Enter the name  
**Basics Html**  
Enter the category  
**Web designing**  
Enter the price  
**700**  
Enter the price limit  
Enter the min limit  
**10000**  
Enter the max limit  
**20000**  
Courses which is in limit 10000 to 20000  
Course 1 Details  
Course Id : 1  
Course Name : Basic java  
Course Category : Java  
Course Price : 15000  
Course 2 Details  
Course Id : 4  
Course Name : C and C++  
Course Category : Basic programs  
Course Price : 13000  
  
  
**Sample Input And Output 2:**  
  
Enter the number of Courses  
**1**  
Course 1 Details  
Enter the id  
**1**  
Enter the name  
**Python Basics**  
Enter the category  
**Python**  
Enter the price  
**2000**  
Enter the price limit  
Enter the min limit  
**4000**  
Enter the max limit  
**6000**  
No course available in this limit

**GroupBy and OrderBy in Linq**

Devin joined as a placement director. He has to analyze all the drives for the academic year 2020 and display the students who’ve got placed in various companies. The student names need to be in ascending order.   
  
Write a program to get the number of students placed ‘n’ and the name of students who have got placed and the company names in which they got placed a list.   
  
**[Note:  Strictly adhere to the object-oriented specifications given as a part of the problem statement.  
Follow the naming conventions as mentioned. Create separate classes in separate files.]**

Consider the class **PlacedStudent** with the following private attributes,

|  |  |
| --- | --- |
| **Data type** | **Attribute** |
| string | \_companyName |
| string | \_studentName |

The methods for **getters**, **setters** and **constructors** are given in the template code.

Consider the class **Program**. It includes the method Main.  
In the Main method,

* read the student details from the user, write a linq query to display the placed students name in ascending order grouped by the company name.

**Note :**

Use GroupBy and OrderBy operator in LINQ.

**Input and Output Format :**  
  
Display the placed students name in **ascending order**.  
Refer sample input and output for formatting specifications.   
**[All text in bold corresponds to input and rest corresponds to output.]**

**Sample Input and Output 1:**  
  
Enter number of companies  
**4**  
Enter company name  
**ABC Tech**  
Enter student name  
**Devin**  
Enter company name  
**BTS**  
Enter student name  
**Peter**  
Enter company name  
**ABC Tech**  
Enter student name  
**Antonio**  
Enter company name  
**BTS**  
Enter student name  
**kanwald**  
Company Name : ABC Tech  
Antonio  
Devin  
Company Name : BTS  
kanwald  
Peter