The following **5 standard LINQ query operators belong to Ordering Operators**category  
OrderBy  
OrderByDescending  
ThenBy  
ThenByDescending  
Reverse   
  
   
  
**OrderBy, OrderByDescending, ThenBy,** and **ThenByDescending**can be used to sort data. **Reverse**method simply reverses the items in a given collection.  
  
We will use the following **Student**class in this demo.

public class Student

{

    public int StudentID { get; set; }

    public string Name { get; set; }

    public int TotalMarks { get; set; }

    public static List<Student> GetAllStudents()

    {

        List<Student> listStudents = new List<Student>

        {

            new Student

            {

                StudentID= 101,

                Name = "Tom",

                TotalMarks = 800

            },

            new Student

            {

                StudentID= 102,

                Name = "Mary",

                TotalMarks = 900

            },

            new Student

            {

                StudentID= 103,

                Name = "Valarie",

                TotalMarks = 800

            },

            new Student

            {

                StudentID= 104,

                Name = "John",

                TotalMarks = 800

            },

        };

        return listStudents;

    }

}

**Example 1:** Sort **Students**by **Name**in **ascending**order

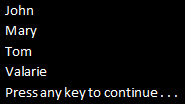
IEnumerable<Student> result = Student.GetAllStudents().OrderBy(s => s.Name);

foreach (Student student in result)

{

    Console.WriteLine(student.Name);

}

**Output:**   
   
  
**Example 2:** Rewrite **Example 1**using **SQL like**syntax

IEnumerable<Student> result = from student in Student.GetAllStudents()

                                                      orderby student.Name

                                                      select student;

foreach (Student student in result)

{

    Console.WriteLine(student.Name);

}

**Output:**  
Same as in **Example 1**  
  
**Example 3:**Sort **Students**by **Name**in **descending**order

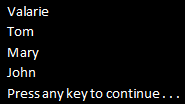
IEnumerable<Student> result = Student.GetAllStudents().OrderByDescending(s => s.Name);

foreach (Student student in result)

{

    Console.WriteLine(student.Name);

}

**Output:**   
   
  
**Example 4:** Rewrite **Example 3** using **SQL like**syntax

IEnumerable<Student> result = from student in Student.GetAllStudents()

                                                      orderby student.Name descending

                                                      select student;

foreach (Student student in result)

{

    Console.WriteLine(student.Name);

}

**Output:**  
Same as in **Example 1**