In this video, we will discuss **different ways of  writing LINQ Queries**.  
  
To write LINQ queries we use the **LINQ Standard Query Operators**. The following are a few Examples of Standard Query Operators  
select  
from  
where   
orderby   
join  
groupby   
  
   
  
**There are 2 ways to write LINQ queries using these Standard Query Operators**  
**1.** **Using Lambda Expressions.** We discussed Lambda Expressions in detail in [Part 99](http://www.youtube.com/watch?v=LDgQ-spnrYY&list=PLAC325451207E3105&index=100)of [C# Tutorial](http://www.youtube.com/playlist?list=PLAC325451207E3105)  
  
**2.** **Using SQL like query expressions**  
  
The **Standard Query Operators**are implemented as extension methods on IEnumerable<T> interface. We will discuss, what extension methods are and how to implement them in a later video session.  
  
**For now let's focus on the 2 ways of writing a LINQ query**. From a performance perspective there is no difference between the two. Which one to use depends on your personal preference. But keep in mind, behind the scene, LINQ queries written using SQL like query expressions are translated into their lambda expressions before they are compiled.   
  
We will use the following Student class in this demo. GetAllStudents() is a static method that returns List<Student>. Since List<T> implements IEnumerable<T>, the LINQ Standard Query Operators will be available and can be applied on List<Student>.

public class Student

{

    public int ID { get; set; }

    public string Name { get; set; }

    public string Gender { get; set; }

    public static List<Student> GetAllStudents()

    {

        List<Student> listStudents = new List<Student>();

        Student student1 = new Student

        {

            ID = 101,

            Name = "Mark",

            Gender = "Male"

        };

        listStudents.Add(student1);

        Student student2 = new Student

        {

            ID = 102,

            Name = "Mary",

            Gender = "Female"

        };

        listStudents.Add(student2);

        Student student3 = new Student

        {

            ID = 103,

            Name = "John",

            Gender = "Male"

        };

        listStudents.Add(student3);

        Student student4 = new Student

        {

            ID = 104,

            Name = "Steve",

            Gender = "Male"

        };

        listStudents.Add(student4);

        Student student5 = new Student

        {

            ID = 105,

            Name = "Pam",

            Gender = "Female"

        };

        listStudents.Add(student5);

        return listStudents;

    }

}

The **LINQ query**should return just the **Male**students.   
  
**LINQ query using Lambda Expressions.**

IEnumerable<Student>students=Student.GetAllStudents**()**

.Where**(**student=>student.Gender=="Male"**);**  
  
**LINQ query using using SQL like query expressions**

IEnumerable<Student>students=fromstudentinStudent.GetAllStudents**()**

wherestudent.Gender=="Male"

selectstudent**;**  
  
**To bind the results of this LINQ query to a GridView**  
GridView1.DataSource = students;  
GridView1.DataBind();