Assignment-2

LINQ Assignment – C#

Code

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Liquery
{

    class Program
    {
        static void Main(string[] args)
        {

            var students = new List<Student>
            {
            new Student { StudentId = 1, Name = "Alice" },
            new Student { StudentId = 2, Name = "Bob" },
            new Student { StudentId = 3, Name = "Charlie" },
            new Student { StudentId = 4, Name = "David" }
            };

            var courses = new List<Course>
            {
            new Course { CourseId = 1, Title = "Math" },
            new Course { CourseId = 2, Title = "Science" },
            new Course { CourseId = 3, Title = "History" }
            };

            var enrollments = new List<Enrollment>
            {
            new Enrollment { StudentId = 1, CourseId = 1 },
            new Enrollment { StudentId = 1, CourseId = 2 },
            new Enrollment { StudentId = 2, CourseId = 2 },
```

```
            new Enrollment { StudentId = 2, CourseId = 3 },
            new Enrollment { StudentId = 3, CourseId = 1 },
            new Enrollment { StudentId = 4, CourseId = 2 }
            };


            Console.WriteLine("List of students enrolled in at least three courses:");

           //  List<Enrollments> s = from en in enrollments
                    //         select en;
           //  foreach(Enrollment o in enrollments )
           //  {
           //      Console.WriteLine(o.StudentId + " " + o.CourseId);
            // }

            var mulresult= students.Join(enrollments, student=> student.StudentId,
               enrollment => enrollment.StudentId,(student,enrollment) => new
{student.StudentId,student.Name}).GroupBy(x => new { x.StudentId, x.Name } )
                .Where(g => g.Count() >= 2)
                .Select(g => g.Key.Name).ToList();

            foreach (var student in mulresult)
            {
               Console.WriteLine(student);
            }

            var studentsGroupedByCourses = students
                    .Select(student => new
                    {
                       student.Name,
                       CourseCount = enrollments.Count(e => e.StudentId ==
student.StudentId)
                    })
                    .GroupBy(s => s.CourseCount)
                    .OrderBy(g => g.Key)
                    .ToList();

                        Console.WriteLine("Group students by the number of courses they are
enrolled in:");
                        foreach (var group in studentsGroupedByCourses)
```

```
                    {
                        var studentNames = string.Join(", ", group.Select(s => s.Name));
                        Console.WriteLine(string.Format("{0} Course(s): {1}", group.Key,
studentNames));
                    }
                    Console.WriteLine();

                    var coursesWithMultipleStudents = enrollments
                    .GroupBy(e => e.CourseId)
                    .Where(g => g.Count() > 1)
                    .Select(g => new
                    {
                        CourseName = courses.First(c => c.CourseId == g.Key).Title,
                        Students = string.Join(", ", g.Join(students, e => e.StudentId, s =>
s.StudentId, (e, s) => s.Name))
                    })
                    .ToList();

                    Console.WriteLine("Courses with students enrolled in more than one
course:");
                    foreach (var course in coursesWithMultipleStudents)
                    {
                        Console.WriteLine(string.Format("Course: {0}, Students: {1}",
course.CourseName, course.Students));
                    }
                    Console.WriteLine();


        var coursesOrderedByEnrollment = enrollments
                .GroupBy(e => e.CourseId)
                .OrderByDescending(g => g.Count())
                .Select(g => new { CourseName = courses.First(c => c.CourseId ==
g.Key).Title, StudentCount = g.Count() })
                .ToList();

        Console.WriteLine("Courses sorted by the number of students enrolled:");

            foreach (var course in coursesOrderedByEnrollment)
            {
```

```
                Console.WriteLine(string.Format("Course: {0}, Students Enrolled:
{1}",course.CourseName,course.StudentCount));
            }



          Console.ReadKey();

       }

   }
}
```

Student.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Liquery
{
   class Student
   {
      public int  StudentId {get; set;}
        public string Name {get; set;}

   }
}
```

Course.cs

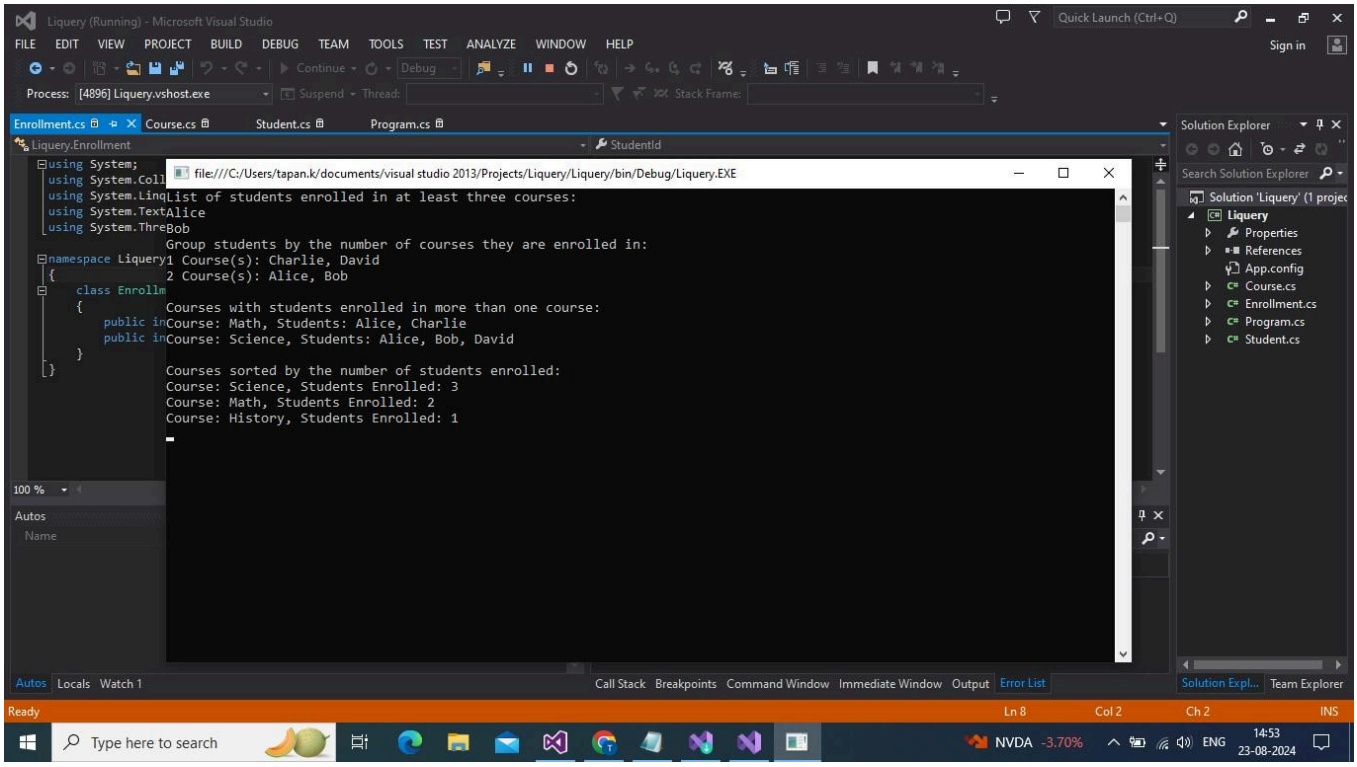
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
using System.Threading.Tasks;

namespace Liquery
{
    class Course
    {
         public int CourseId {get; set;}
         public string Title { get; set; }
    }
}
```

Enrollment.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Liquery
{
    class Enrollment
    {
       public int StudentId  { get; set; }
       public int CourseId {get; set;}
    }
}
```

Output:


Liquery (Running) - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP
Sign in
Process: [4896] Liquery.vshost.exe
Enrollment.cs Course.cs Student.cs Program.cs
Liquery.Enrollment
StudentId
file:///C:/Users/tapan.k/documents/visual studio 2013/Projects/Liquery/Liquery/bin/Debug/Liquery.EXE
List of students enrolled in at least three courses:
Alice
Bob
Group students by the number of courses they are enrolled in:
1 Course(s): Charlie, David
2 Course(s): Alice, Bob
Courses with students enrolled in more than one course:
Course: Math, Students: Alice, Charlie
Course: Science, Students: Alice, Bob, David
Courses sorted by the number of students enrolled:
Course: Science, Students Enrolled: 3
Course: Math, Students Enrolled: 2
Course: History, Students Enrolled: 1
Solution Explorer
Solution 'Liquery' (1 projec
Liquery
Properties
References
App.config
Course.cs
Enrollment.cs
Program.cs
Student.cs
Autos
Name
Autos Locals Watch 1
Call Stack Breakpoints Command Window Immediate Window Output Error List
Solution Expl... Team Explorer
Ready
Ln 8 Col 2 Ch 2 INS
Type here to search
NVDA -3.70%
ENG
14:53
23-08-2024