Name: Sownthari R P

Date: 22.08.2024

1.LINQ:

Student.cs

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

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

    }
}
```

2.Courses.cs

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

namespace LinqAssignment
{
    internal class Course
```

```
    {
        public int CourseId { get; set; }
        public string Title { get; set; }

    }
}
```

3.Enrollment.cs

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

namespace LinqAssignment
{
    internal class Enrollment
    {
        public int StudentId { get; set; }
        public int CourseId { get; set; }

    }
}
```

4.Program.cs

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

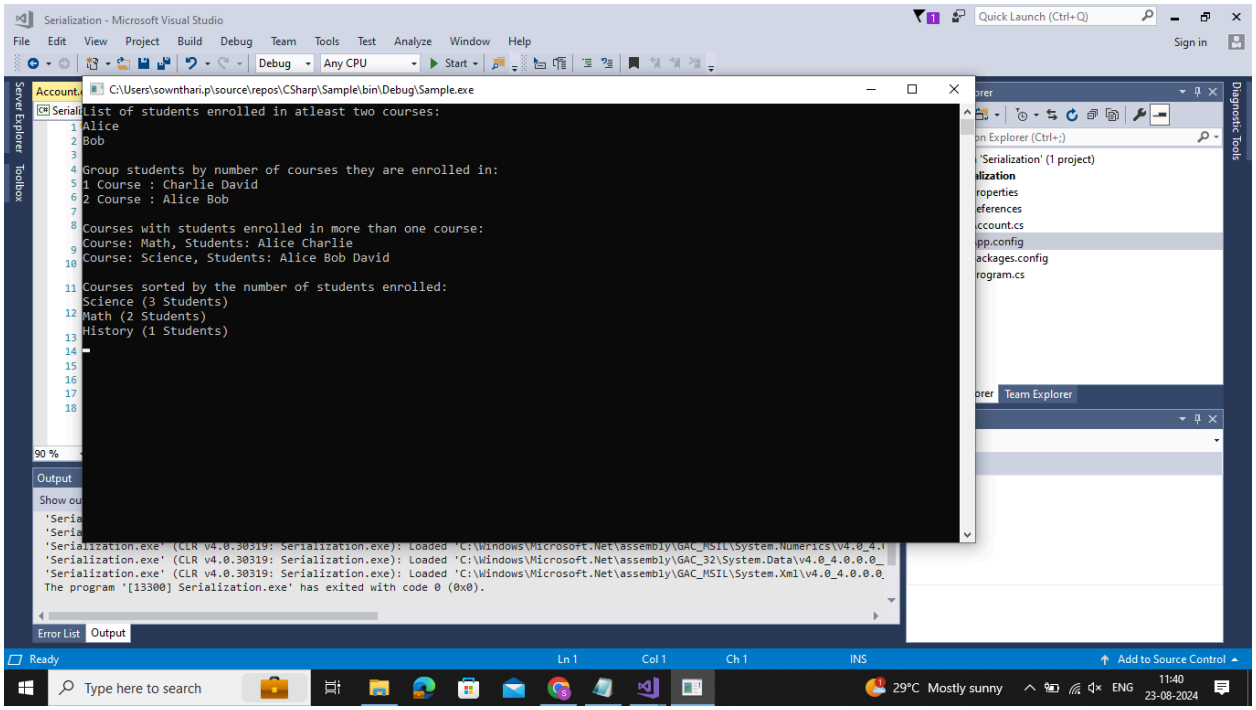
```
namespace LinqAssignment
{
    class Program
    {
        static void Main(string[] args)
        {
            List<Student> students = new List<Student>()
            {
                new Student { StudentId = 1, Name = "Alice" },
                new Student { StudentId = 2, Name = "Bob" },
                new Student { StudentId = 3, Name = "Charlie" },
                new Student { StudentId = 4, Name = "David" }
            };

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

            List<Enrollment> 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 }
            };
```

```
        IEnumerable<string>  result = (students.Where(student =>
enrollments.Count(enroll => enroll.StudentId == student.StudentId) >= 2)
            .Select(student => student.Name))
            .ToList();

        Console.WriteLine("List of students enrolled in atleast two courses:");

        foreach(string name in result)
        {
            Console.WriteLine(name);
        }
        Console.WriteLine();
        Console.WriteLine("Group students by number of courses they are enrolled in:
");

        var r = enrollments.GroupBy(enrollment => enrollment.StudentId).GroupBy(c =>
c.Count()).OrderBy(c => c.Key).ToList();

        foreach (var group in r)
        {
            Console.Write(group.Key+" Course : ");
            foreach (var e in group)
            {
                var s = students.Join(e, st => st.StudentId, en => en.StudentId, (st, en) =>
new
                {
                    studentName = st.Name
                }).Distinct();
                foreach(var a in s)
                {
                    Console.Write(a.studentName+" ");
```

```
            }

        }
        Console.WriteLine();
    }



    var list = courses.Where(c => enrollments.Count(e => e.CourseId ==
c.CourseId) > 1)
        .Select(c => new
        {
            CourseTitle = c.Title,
            Students = enrollments.Where(e => e.CourseId == c.CourseId)
                        .Select(e => students.First(s => s.StudentId ==
e.StudentId).Name)
                        .Distinct()
                        .ToList()
        })
        .ToList();
    Console.WriteLine();
    Console.WriteLine("Courses with students enrolled in more than one course: ");
    foreach (var course in list)
    {
        Console.Write("Course: " + course.CourseTitle + ", Students: ");
        foreach (var student in course.Students)
        {
            Console.Write(student + " ");
        }
        Console.WriteLine();
```

```
        }
        Console.WriteLine();
        Console.WriteLine("Courses sorted by the number of students enrolled: ");
        var sort = enrollments.GroupBy(enrollment =>
enrollment.CourseId).OrderByDescending(c => c.Count());

        foreach (var group in sort)
        {
            var c = courses.Where(course => course.CourseId ==
group.Key).Select(course => course.Title).ToList();
            foreach (var course in c)
            {
                int count = group.Count();
                Console.WriteLine($"{course} ({count} Students)");
            }
        }

        Console.ReadKey();
    }
  }
}
```

Output:


Serialization - Microsoft Visual Studio
C:\Users\sownthari.p\source\repos\CSharp\Sample\bin\Debug\Sample.exe
List of students enrolled in atleast two courses:
Alice
Bob
Group students by number of courses they are enrolled in:
1 Course : Charlie David
2 Course : 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:
Science (3 Students)
Math (2 Students)
History (1 Students)
The program '[13300] Serialization.exe' has exited with code 0 (0x0).
Ready
Type here to search
29°C Mostly sunny
ENG
11:40
23-08-2024