DATE : 22/08/2024

LINQ Assignment – C#

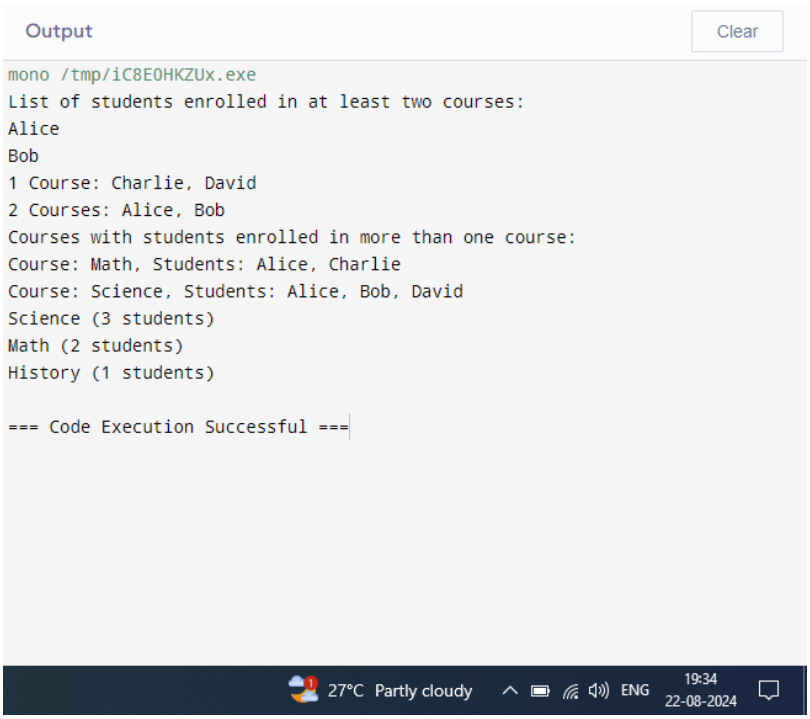
CODE:

```
using System;
using System.Linq;
using System.Collections.Generic;
public class Student{
   public int StudentId{get; set;}
   public string Name{get; set;}
}
public class Course{
   public int CourseId{get; set;}
   public string Title{get; set;}
}
public class Enrollment{
    public int StudentId{get; set;}
    public int CourseId{get; set;}
}
public class Program
{
   public 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 two courses:");
    var result = from e in enrollments
                 group e by e.StudentId into g
                 where g.Count() >= 2
                 select g.Key;

    IList<string> studentNames = new List<string>();

    foreach (var studentId in result)
    {
      string studentName = (from p in students
                            where p.StudentId == studentId
                            select p.Name).First();

      studentNames.Add(studentName);
    }

    foreach (string name in studentNames)
    {
      Console.WriteLine(name);
    }

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

foreach (var group in studentCourseGroups)
{
  Console.WriteLine($"{group.Key} Course{(group.Key > 1 ? "s" : "")}: {string.Join(", ",
group.Select(g => g.Name))}");
}


var coursesWithMultipleStudents = from enrollment in enrollments
                                  join student in students on enrollment.StudentId equals
student.StudentId
```

```
                        group new { student.Name, enrollment.CourseId } by
enrollment.CourseId into courseGroup
                        where courseGroup.Count() > 1
                        join course in courses on courseGroup.Key equals course.CourseId
                        select new
                        {
                            CourseTitle = course.Title,
                            Students = courseGroup.Select(cg => cg.Name).Distinct()
                        };

        Console.WriteLine("Courses with students enrolled in more than one course:");
        foreach (var course in coursesWithMultipleStudents)
        {
            Console.WriteLine($"Course: {course.CourseTitle}, Students: {string.Join(", ",
course.Students)}");
        }

var courseEnrollmentCounts = enrollments
    .GroupBy(e => e.CourseId)
    .Select(g => new
    {
        CourseId = g.Key,
        StudentCount = g.Select(e => e.StudentId).Distinct().Count()
    })
    .Join(courses, ec => ec.CourseId, c => c.CourseId, (ec, cr) => new
    {
        CourseTitle = cr.Title,
        StudentCount = ec.StudentCount
    })
    .OrderByDescending(c => c.StudentCount)
    .ToList();

foreach (var course in courseEnrollmentCounts)
{
    Console.WriteLine($"{course.CourseTitle} ({course.StudentCount} students)");
}

    }
}
```

OUTPUT:


Output
Clear
mono /tmp/iC8EOHKZUx.exe
List of students enrolled in at least two courses:
Alice
Bob
1 Course: Charlie, David
2 Courses: Alice, Bob
Courses with students enrolled in more than one course:
Course: Math, Students: Alice, Charlie
Course: Science, Students: Alice, Bob, David
Science (3 students)
Math (2 students)
History (1 students)
=== Code Execution Successful ===
27°C Partly cloudy
ENG
19:34
22-08-2024