

Day 3

Name : DHANAPAL

Date : 23/08/2024

Code :

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

```
namespace Day4Assignment
{
    class Student
    {
        public int StudentId { get; set; }
        public string Name { get; set; }
    }
    class Course
    {
        public int CourseId { get; set; }
        public string Title { get; set; }
    }
    class Enrollment
    {
        public int StudentId { get; set; }
        public int CourseId { get; set; }
    }
    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 }
    };

    // select name from students where studentid in(select studentid from
    enrollments groupby studentid having count(studentid)>=2)
    var q1 = students.Where(s => enrollments.Count(e => e.StudentId ==
    s.StudentId) >= 2)
        .Select(s => s.Name)
        .ToList();

    Console.WriteLine("Students enrolled in at least two courses:");
    foreach (var student in q1)
    {
        Console.WriteLine(student);
    }
    Console.WriteLine();
    var q2 = students.GroupBy(s => enrollments.Count(e => e.StudentId ==
    s.StudentId))
        .Select(g => new { CourseCount = g.Key, Students = g.Select(s =>
    s.Name).ToList() })
        .ToList();

```

```

Console.WriteLine("Students grouped by the number of courses:");
foreach (var group in q2)
{
    Console.Write("Course Count: " + group.CourseCount + ", Students: ");
    foreach (var student in group.Students)
    {
        Console.Write(student + " ");
    }
    Console.WriteLine();
}
Console.WriteLine();
var q3 = 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 more than one student enrolled:");
foreach (var course in q3)
{
    Console.Write("Course: " + course.CourseTitle + ", Students: ");
    foreach (var student in course.Students)
    {
        Console.Write(student + " ");
    }
    Console.WriteLine();
}
Console.WriteLine();
var q4 = courses.Select(c => new
{

```

```

        CourseTitle = c.Title,
        StudentCount = enrollments.Count(e => e.CourseId == c.CourseId)
    })
    .OrderByDescending(c => c.StudentCount)
    .ToList();

    Console.WriteLine("Courses sorted by the number of students enrolled:");
    foreach (var course in q4)
    {
        Console.WriteLine(course.CourseTitle + " (" + course.StudentCount + "
students)");
    }
    Console.ReadKey();
}
}
}
}
}

```

Output :

