DAY 4:
Name :Siddarth S
Date: 22-08-2024

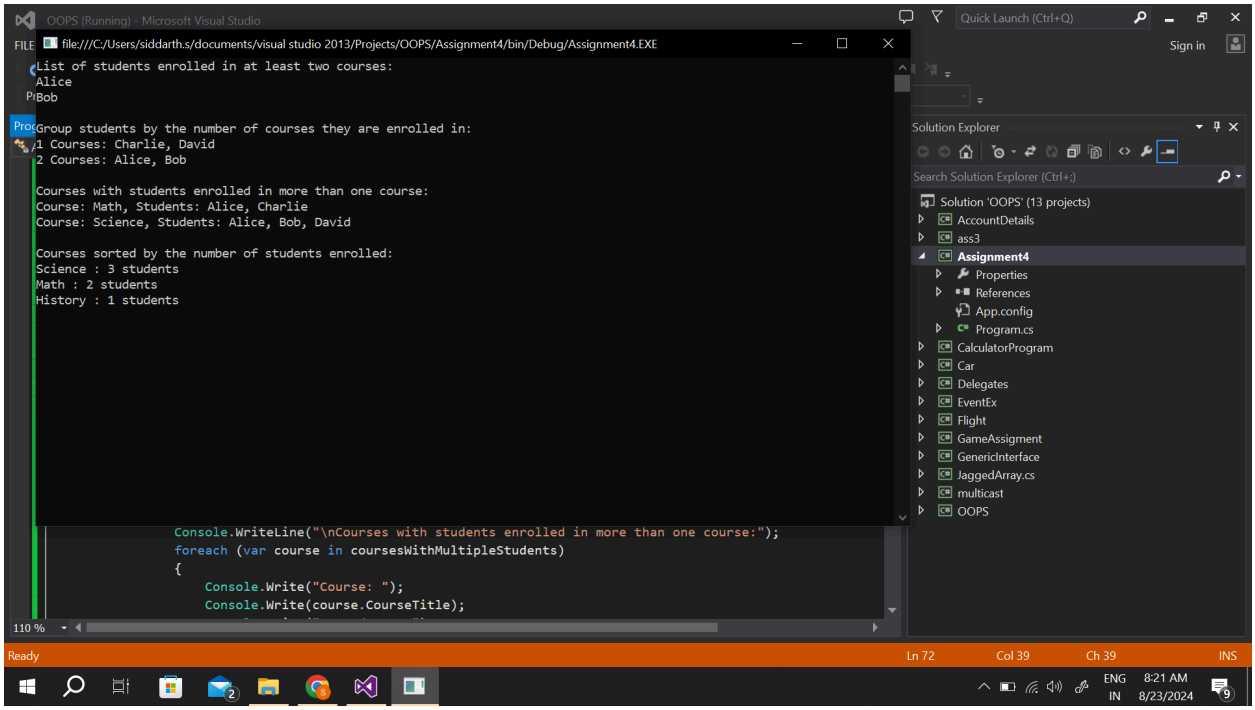
Program.cs

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

namespace Assignment4
{

    class Student
    {
        internal int StudentId { get; set; }
        internal string Name { get; set; }
    }
    class Course
    {
        internal int CourseId { get; set; }
        internal string Title { get; set; }
    }
    class Enrollment
    {
        internal int StudentId { get; set; }
        internal 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 }
            };

            var studentsWithAtLeastTwoCourses = students
             .Where(s => enrollments.Count(e => e.StudentId == s.StudentId) >= 2)
             .ToList();

            Console.WriteLine("List of students enrolled in at least two courses:");
            foreach (var student in studentsWithAtLeastTwoCourses)
            {
                Console.WriteLine(student.Name);
            }

           var studentsGroupedByCourses =
             (from s in students
              group s by (from e in enrollments where e.StudentId == s.StudentId select e).Count()
into g
              orderby g.Key
              select g).ToList();

            Console.WriteLine("\nGroup students by the number of courses they are enrolled in:");
            foreach (var group in studentsGroupedByCourses)
            {
                Console.Write(group.Key);
                Console.Write(" Courses: ");
                Console.WriteLine(string.Join(", ", group.Select(s => s.Name)));
            }

            var coursesWithMultipleStudents =
                (from c in courses
                 join e in enrollments on c.CourseId equals e.CourseId
                 group new { c.Title, e.StudentId } by c.Title into g
                 where g.Select(ce => ce.StudentId).Distinct().Count() > 1
                 select new
                 {
```

```
                CourseTitle = g.Key,
                Students = (from g2 in g
                            join s in students on g2.StudentId equals s.StudentId
                            select s.Name).Distinct()
            }).ToList();

            Console.WriteLine("\nCourses with students enrolled in more than one course:");
            foreach (var course in coursesWithMultipleStudents)
            {
                Console.Write("Course: ");
                Console.Write(course.CourseTitle);
                Console.Write(", Students: ");
                Console.WriteLine(string.Join(", ", course.Students));
            }
            var coursesSortedByEnrollment =
                (from c in courses
                 join e in enrollments on c.CourseId equals e.CourseId into g
                 let StudentCount = g.Count()
                 orderby StudentCount descending
                 select new
                 {
                     CourseTitle = c.Title,
                     StudentCount = StudentCount
                 }).ToList();

            Console.WriteLine("\nCourses sorted by the number of students enrolled:");
            foreach (var course in coursesSortedByEnrollment)
            {
                Console.Write(course.CourseTitle);
                Console.Write(" : ");
                Console.Write(course.StudentCount);
                Console.WriteLine(" students");
            }


        Console.ReadKey();

    }
  }
}
```

Output:


OOPS (Running) - Microsoft Visual Studio
file:///C:/Users/siddarth.s/documents/visual studio 2013/Projects/OOPS/Assignment4/bin/Debug/Assignment4.EXE
List of students enrolled in at least two courses:
Alice
Bob
Group students by the number of courses they are enrolled in:
1 Courses: 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
Courses sorted by the number of students enrolled:
Science : 3 students
Math : 2 students
History : 1 students
Console.WriteLine("\nCourses with students enrolled in more than one course:");
foreach (var course in coursesWithMultipleStudents)
{
Console.Write("Course: ");
Console.Write(course.CourseTitle);
Solution Explorer
Solution 'OOPS' (13 projects)
AccountDetails
ass3
Assignment4
Properties
References
App.config
Program.cs
CalculatorProgram
Car
Delegates
EventEx
Flight
GameAssigment
GenericInterface
JaggedArray.cs
multicast
OOPS
Ready
Ln 72
Col 39
Ch 39
INS
8:21 AM
8/23/2024