

DAY 4

C#

LINQ

1. Filter: Retrieve a list of students who are enrolled in at least two courses.

CODE:

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

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

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

public class Enrollment{
    public int StudentId {set; get;}
    public int CourseId {set; get;}
}

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 }
};
```

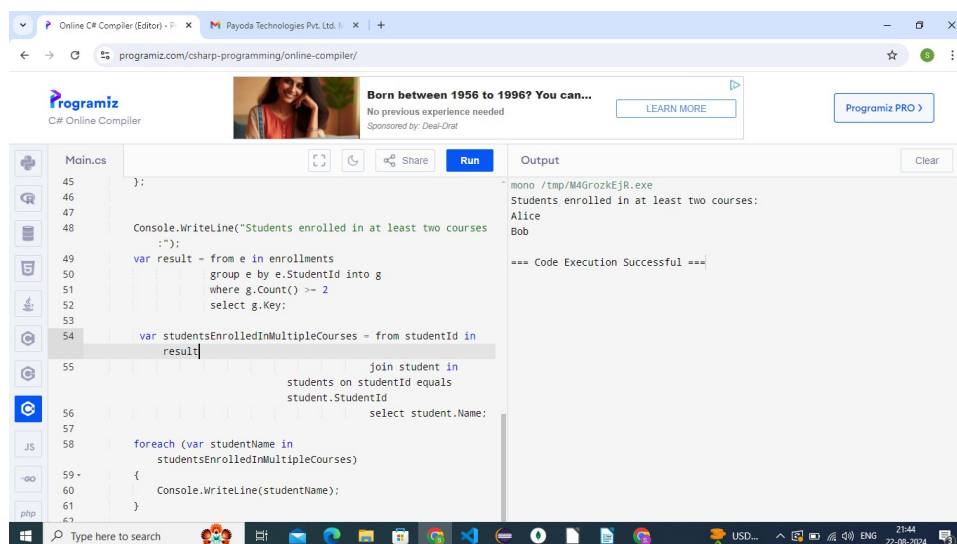
```
Console.WriteLine("Students enrolled in at least two courses:");
var result = from e in enrollments
              group e by e.StudentId into g
              where g.Count() >= 2
              select g.Key;
```

```
var studentsEnrolledInMultipleCourses = from studentId in result
                                          join student in students on studentId equals student.StudentId
                                          select student.Name;
```

```
foreach (var studentName in studentsEnrolledInMultipleCourses)
{
    Console.WriteLine(studentName);
}

}
```

OUTPUT:



The screenshot shows the Programiz Online C# Compiler interface. The code editor on the left contains the following C# code:

```

45  };
46
47  Console.WriteLine("Students enrolled in at least two courses
48  :");
49  var result = from e in enrollments
50                group e by e.StudentId into g
51                where g.Count() >= 2
52                select g.Key;
53
54  var studentsEnrolledInMultipleCourses = from studentId in
55                                          result
56                                          join student in
57                                          students on studentId equals
58                                          student.StudentId
59                                          select student.Name;
60
61  foreach (var studentName in
62          studentsEnrolledInMultipleCourses)
63  {
64      Console.WriteLine(studentName);
65  }
66
67  }
```

The output window on the right shows the following output:

```

mono /tmp/M4GrozKEjR.exe
Students enrolled in at least two courses:
Alice
Bob
=== Code Execution Successful ===
```

2. Group: Group the students by the number of courses they are enrolled in.

CODE:

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

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

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

public class Enrollment
{
    public int StudentId { set; get; }
    public int CourseId { set; get; }
}

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("Grouping students by the number of courses they are enrolled
in:");

    var groupedStudents = from e in enrollments
        group e by e.StudentId into g
        join student in students on g.Key equals student.StudentId
        group student by g.Count() into grouped
        orderby grouped.Key
        select grouped;

    foreach (var group in groupedStudents)
    {
        Console.WriteLine($"Number of Courses: {group.Key}");
        foreach (var student in group)
        {
            Console.WriteLine($" {student.Name}");
        }
    }
}

```

OUTPUT:

The screenshot shows a web browser window with the URL `programiz.com/csharp-programming/online-compiler/`. The page features a banner for "Programiz" with a logo and a text box that says "Born between 1956 to 1996? You can..." with a "LEARN MORE" button. Below the banner, there is a "Main.cs" file editor with the following C# code:

```

53 courses they are enrolled in:");
54 var groupedStudents = from e in enrollments
55     group e by e.StudentId into g
56     join student in students on g.Key
57     equals student.StudentId
58     group student by g.Count() into
59     grouped
60     orderby grouped.Key
61     select grouped;
62
63 foreach (var group in groupedStudents)
64 {
65     Console.WriteLine($"Number of Courses: {group.Key}");
66     foreach (var student in group)
67     {
68         Console.WriteLine($" {student.Name}");
69     }
70 }
71

```

The "Output" pane on the right shows the following text:

```

mono /tmp/JZzqV9XrmS.exe
Grouping students by the number of courses they are enrolled in:
Number of Courses: 1
Charlie
David
Number of Courses: 2
Alice
Bob
--- Code Execution Successful ---

```

The bottom of the browser window shows a Windows taskbar with a search bar, several application icons, and a system tray displaying the temperature (27°C), time (21:47), and date (22-08-2024).

3. Join: Find the list of courses that have students enrolled in more than one course, along with the names of those students.

CODE:

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

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

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

public class Enrollment
{
    public int StudentId { set; get; }
    public int CourseId { set; get; }
}

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("Courses with students enrolled in more than one course, along
with their names:");

```

```

var result = from e in enrollments
              group e by e.StudentId into g
              where g.Count() > 1
              from enrollment in g
              join student in students on enrollment.StudentId equals student.StudentId
              join course in courses on enrollment.CourseId equals course.CourseId
              select new { CourseTitle = course.Title, StudentName = student.Name };

```

```

var groupedResults = from r in result
                      group r by r.CourseTitle into g
                      select g;

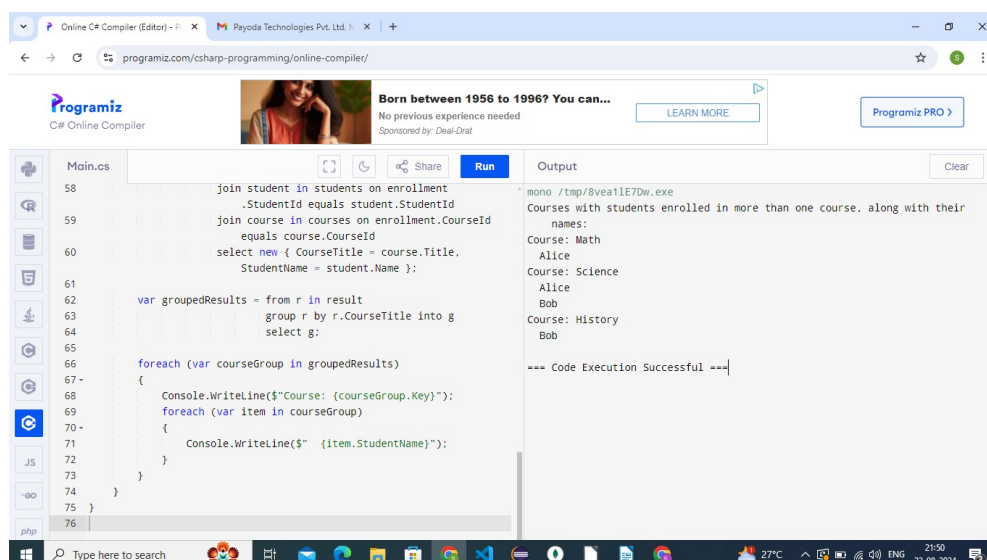
```

```

foreach (var courseGroup in groupedResults)
{
    Console.WriteLine($"Course: {courseGroup.Key}");
    foreach (var item in courseGroup)
    {
        Console.WriteLine($" {item.StudentName}");
    }
}
}
}

```

CODE:



The screenshot shows the Programiz Online C# Compiler interface. The code in the editor is as follows:

```

Main.cs
58      join student in students on enrollment
59      .StudentId equals student.StudentId
60      join course in courses on enrollment.CourseId
61      equals course.CourseId
62      select new { CourseTitle = course.Title,
63                  StudentName = student.Name };
64
65
66      var groupedResults = from r in result
67                          group r by r.CourseTitle into g
68                          select g;
69
70      foreach (var courseGroup in groupedResults)
71      {
72          Console.WriteLine($"Course: {courseGroup.Key}");
73          foreach (var item in courseGroup)
74          {
75              Console.WriteLine($" {item.StudentName}");
76          }
77      }

```

The output window shows the following results:

```

mono /tmp/8veat1E7Dw.exe
Courses with students enrolled in more than one course, along with their
names:
Course: Math
Alice
Course: Science
Alice
Bob
Course: History
Bob
=== Code Execution Successful ===

```

4. Order: Sort the courses by the number of students enrolled in descending order.

CODE:

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

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

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

public class Enrollment
{
    public int StudentId { set; get; }
    public int CourseId { set; get; }
}

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 }
    };

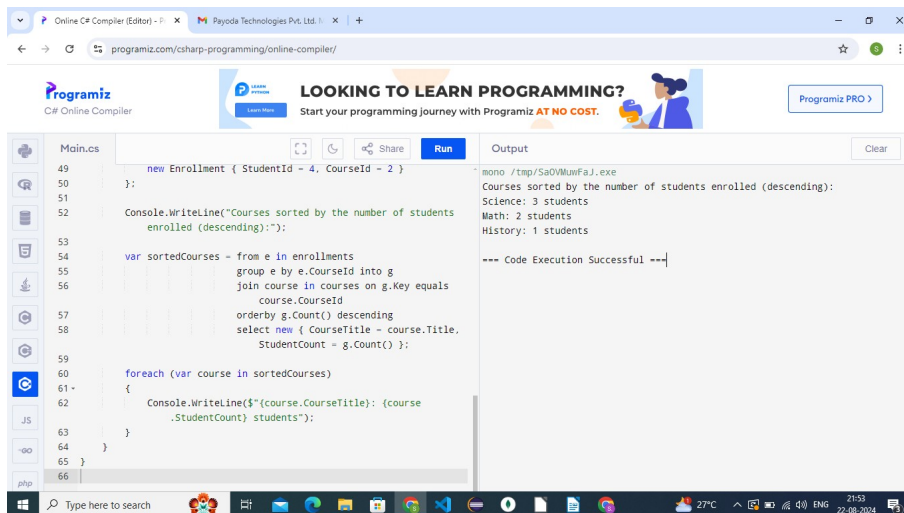
    Console.WriteLine("Courses sorted by the number of students enrolled (descending):");

    var sortedCourses = from e in enrollments
                        group e by e.CourseId into g
                        join course in courses on g.Key equals course.CourseId
                        orderby g.Count() descending
                        select new { CourseTitle = course.Title, StudentCount = g.Count() };

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

```

OUTPUT:



The screenshot shows the Programiz C# Online Compiler interface. The code editor on the left contains the following C# code:

```

49 new Enrollment { StudentId = 4, CourseId = 2 }
50 };
51
52 Console.WriteLine("Courses sorted by the number of students
53 enrolled (descending):");
54
55 var sortedCourses = from e in enrollments
56                     group e by e.CourseId into g
57                     join course in courses on g.Key equals
58                     course.CourseId
59                     orderby g.Count() descending
60                     select new { CourseTitle = course.Title,
61                                 StudentCount = g.Count() };
62
63 foreach (var course in sortedCourses)
64 {
65     Console.WriteLine($"{course.CourseTitle}: {course
66     .StudentCount} students");
67 }

```

The output window on the right shows the following output:

```

mono /tmp/SaOVMunFaj.exe
Courses sorted by the number of students enrolled (descending):
Science: 3 students
Math: 2 students
History: 1 students
--- Code Execution Successful ---

```