Classroom

Preparation

Download the skeleton provided in Judge. **Do not** change the **StartUp** class or its **namespace**.

Problem description

Your task is to create a repository, which stores items by creating the classes described below.

First, write a **C#** class **Student** with the following properties:

- **FirstName: string**
- **LastName: string**
- **Subject: string**

The class **constructor** should receive **firstName, lastName** and **subject.** You need to create the appropriate **getters and setters**. The class should override the **ToString()** method in the following format:

```
"Student: First Name = {firstName}, Last Name = {lastName}, Subject = {subject}"
```

Next, write a **C#** class **Classroom** that has **students** (a collection, which stores the **students**) and a certain capacity. All entities inside the repository have the **same fields**. Also, the **Classroom** class should have the following properties:

- **Capacity: int**
- **Count: int – returns the number of students in the classroom**

The class **constructor** should receive **capacity**, also it should initialize the **students** with a new instance of the collection. Implement the following features:

- Field **students** – **collection** that holds added students
- Method **RegisterStudent(Student student)** – **adds** an **entity** to the students **if there is** an **empty seat** for the student.
    - Returns **"Added student {firstName} {lastName}"** if the student is successfully added
    - Returns **"No seats in the classroom" –** if there are no more seats in the classroom

- Method **DismissStudent(string firstName, string lastName)** – removes the student by **the given names**
    - Returns **"Dismissed student {firstName} {lastName}"** if the student is successfully dismissed
    - Returns **"Student not found"** if the student is not in the classroom

- Method **GetSubjectInfo(string subject)** – returns all the students with the **given subject in the following format:**

```
"Subject: {subjectName}
Students:
{firstName} {lastName}
{firstName} {lastName}
…"
```

    - Returns **"No students enrolled for the subject"** if the student is not in the classroom

- Method **`GetStudentsCount()`** – returns the **count** of the **students in the classroom**.
- Method **`GetStudent(string firstName, string lastName)`** – returns the student with the given names.

Constraints

- The **combinations** of **names** will **always be unique**.
- The **capacity** will always be **a positive number**.

Examples

This is an example of how the **`Classroom`** class is **intended to be used**.

Sample code usage

```
// Initialize the repository
Classroom classroom = new Classroom(10);
// Initialize entities
Student student = new Student("Peter", "Parker", "Geometry");
Student studentTwo = new Student("Sarah", "Smith", "Algebra");
Student studentThree = new Student("Sam", "Winchester", "Algebra");
Student studentFour = new Student("Dean", "Winchester", "Music");
// Print Student
Console.WriteLine(student); // Student: First Name = Peter, Last Name = Parker, Subject =
Geometry
// Register Student
string register = classroom.RegisterStudent(student);
Console.WriteLine(register); // Added student Peter Parker
string registerTwo = classroom.RegisterStudent(studentTwo);
string registerThree = classroom.RegisterStudent(studentThree);
string registerFour = classroom.RegisterStudent(studentFour);
// Dismiss Student
string dismissed = classroom.DismissStudent("Peter", "Parker");
Console.WriteLine(dismissed); // Dismissed student Peter Parker
string dismissedTwo = classroom.DismissStudent("Ellie", "Goulding");
Console.WriteLine(dismissedTwo); // Student not found
// Subject info
string subjectInfo = classroom.GetSubjectInfo("Algebra");
Console.WriteLine(subjectInfo);
// Subject: Algebra
// Students:
// Sarah Smith
// Sam Winchester
string anotherInfo = classroom.GetSubjectInfo("Art");
Console.WriteLine(anotherInfo); // No students enrolled for the subject
// Get Student
```

```
Console.WriteLine(classroom.GetStudent("Dean", "Winchester"));
// Student: First Name = Dean, Last Name = Winchester, Subject = Music
```

Submission

Zip all the files in the project folder except **bin** and **obj** folders

Submit **single .zip file**, containing **`Classroom` package, with the classes inside (`Student`, `Classroom` and the `StartUp` class)**. There is no specific content required inside the **`StartUp`** class e. g. you can do any kind of local testing of you program there. However there should be **`Main(string[] args)`** method inside.