

# Programming in C#

## Starter Assignment 1

### Learning outcomes

0. Basic understanding of Objects.
1. Style & Maintainability
2. Documentation

### Project course administration

Create a project course administration application, based on the description below.

#### Case description

Create a simple project course administration application for a teacher. This application should enable a user to add and search for student(s) and/or project group(s). This should be done via a GUI or Console application.

It should support the following operations

1. Adding a student.
2. View all students. (ListBox or printout)
3. View all groups.
4. Search students by number.
5. Show all students in a group.
6. Show statistics.

#### User interaction

A user should be able to do the following things:

1. Add a student:
  - The user provides the required information and confirms.
  - Information to record: name and student number.
  - The students should be added only when the student number is unique.
  - When the student is added, he/she also needs to automatically be assigned to a project group. When the existing group(s) are full, a new group is created. o When a student has been added, all the students are shown in the ListBox/printed.
2. View all students:
  - Display the students.
3. View all groups:
  - Display the groups.
4. Search for a student:
  - The user provides a student number and confirms.
  - The student with the student number is displayed.
5. Show all students in a group:

- The user provides a group name and confirms.
  - Display all students belonging to that group.
6. Show statistics: The course name, amount of students and amount of groups.

## UML

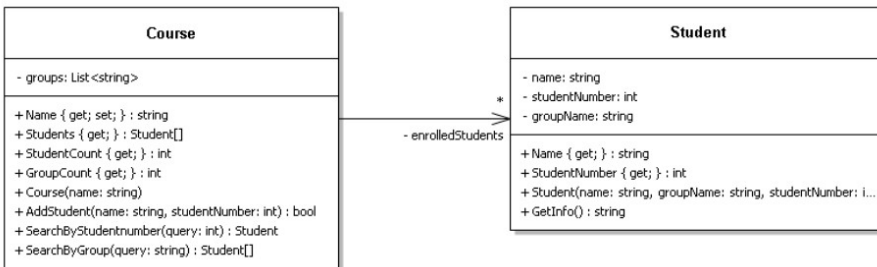


Figure 1: Class Diagram

Course Class	
Properties	Description
Students	Should return all the added students.
StudentCount	Should return the number of students added to the system. Note: This property does not have a backing field!
GroupCount	Should return the number of groups created by the system. Note: This property does not have a backing field!
Constructors & methods	Description
AddStudent	<p>Add a student to the application. Also implement code to automatically assign a student to a group. This is achieved by determining if a group is full or not; a group can have no more than 3 students:</p> <ul style="list-style-type: none"> <li>- When the group is not full: create the Student-object with the available groupName.</li> <li>- When the group is full: create a new groupName, add it to the groups-list and create the Student-object with the new groupName. The groupName is a string generated in the form PG. <i>For example: PG1_PCS2, PG2_J PCS2, PG3 PCS2, etc.</i></li> </ul>
SearchByStudentNumber	Return a student with the student number (e.g. query-parameter), if there is none return null.

SearchByGroup                      Return all students in a group  
(e.g. query-parameter), if there is no group found an  
empty array should be returned. Note that it is not  
possible to access the groupName in the Student class.  
Decide for yourself how you can best do this, while  
making sure the groupName can only be set on  
Student-object construction.

---

---

#### Student Class

---

Instance variable	Description
Name	The name of the student. Make sure this can never be assigned an empty string to it.
studentNumber	The student number. Make sure this is a number >= 10000 or -1
Constructors & methods	Description
Student	This should initialize all the instance variable.
GetInfo	Should return a string to show in a control. The format of the string should be: <name> (<student number>) - <groupName>

---

**Note: you are NOT allowed to change the existing fields methods and you must use them whenever possible. If extra methods are needed you may add them but they should not duplicate existing functionality.**

#### Before delivering your work:

- Test if it builds
- Check if all functionality is in your delivery.
- Make sure the interfaces are correct. Check with the UML-diagram.
- Test functionality