**Academy.Models.Course class**

* **Constructor()**
  + Should correctly assign passed values.
  + Should correctly initialize the collections.
* **Name**
  + Should throw ArgumentException when passed value is invalid.
  + Should not throw when the passed value is valid.
  + Should correctly assign passed value.
* **LecturesPerWeek**
  + Should throw ArgumentException when passed value is invalid.
  + Should not throw when the passed value is valid.
  + Should correctly assign passed value.
* **StartingDate**
  + Should throw ArgumentNullException when passed value is invalid.
  + Should not throw when the passed value is valid.
  + Should correctly assign passed value.
* **EndingDate**
  + Should throw ArgumentNullException when passed value is invalid.
  + Should not throw when the passed value is valid.
  + Should correctly assign passed value.
* **ToString()**
  + Should return passed data a list of lectures.
    - **Hint**: Only check if the method iterates over the collections, do not check the format of the users.
  + Should return passed data and a message saying there are no lectures in the course.

**Academy.Models.Season class**

* **ListUsers()**
  + Should return a list of students and trainers.
    - **Hint**: Only check if the method iterates over the collections, do not check the format of the users.
  + Should return message saying that there are no users in this season.
* **ListCourses()**
  + Should return a list of courses.
    - **Hint**: Only check if the method iterates over the collection, do not check the format of the courses.
  + Should return message saying that there are no courses in this season.

**Academy.Models.Abstractions.User class**

* **Constructor()**
  + Should correctly assign passed values.
* **Username**
  + Should throw ArgumentException when passed value is invalid.
  + Should not throw when the passed value is valid.
  + Should correctly assign passed value.
* **Hint**: How can we create an instance of an abstract class?

**Academy.Core.Factories.AcademyFactory class**

* **CreateLectureResource()**
  + Should throw ArgumentException when passed type is invalid.
  + Should return correct instances with correctly assigned data.

**Academy.Commands.Adding.AddStudentToSeasonCommand class**

* **Constructor()**
  + Should throw ArgumentNullException when a passed provider is null.
  + Should correctly assign passed values.
    - **Hint**: How do we access private fields? Expose.
* **Execute()**
  + Should throw ArgumentException when the passed student is already a part of that season.
  + Should correctly add the found student into the season.
  + Should return a success message that contains the student's username and season ID.
* **Hint**: How can we create an instance of an internal class?

**Academy.Commands.Adding.AddStudentToCourseCommand class**

* **Constructor()**
  + Should throw ArgumentNullException when a passed provider is null.
  + Should correctly assign passed values.
* **Execute()**
  + Should throw ArgumentException when the passed course form is invalid.
  + Should correctly add the found student into the course in the coresponding attendance form.
  + Should return a success message that contains the student's username and season ID.