PBL: Grade Book Part 8: `class AssignmentLoader`

©2026 Chris Nielsen – *www.nielsenedu.com*

This assignment continues the project to create software for a school grade book that will hold the scores that students received for their assignments.

This part of the assignment will implement `class AssignmentLoader` that will load a file that contains both the information about an assignment to create an `Assignment` object, and the score each student has received on the assignment.

The format of the file is as follows:

- The first line of text contains the assignment name.
- The second line of text contains the maximum score for the assignment as an integer, then optionally white space followed by the word "`strict`" if the student will not be able to receive a score higher than the maximum score.
- Each subsequent line is to contain the student's student number and score, separated by a comma. The remainder of the line can contain any data, such as the student's name or a comment about the assignment.

`Scores_a1.txt`

```
A1 - Primitive Types 1
10 strict
1234,10
2345,5,Huang,James J. – James was feeling ill this day.
3456,10 Long,Chen
4567,10,Zhuge,Liang
```

Implement class AssignmentLoader according to the following specification.

- The class is to contain at least these three fields:
  - A field of type `String` to store the filename where the assignment information has been saved.
  - A field of type `StudentList` to store the list of students – this is needed so that each score can be added to the appropriate student's existing assignment scores.
  - A field of type `Assignment` that will store the `Assignment` object that will be created using the details about the assignment as loaded from the file.
- A constructor that takes two parameters:
  - the filename to save in the filename field.
  - the list of students to store in the `StudentList` field.
- A load method that will load all the data from the file specified by the filename field. It must:
  - Create an assignment object and save that in the `Assignment` field.
  - Load all student scores, and for every score successfully loaded, save that score into the `Student` object it belongs to. *Hint*: the student number will be used to determine which `Student` object the score should be added to. You may wish to refer to the structural diagram in the previous part of the assignment to aid in understanding how `AssignmentScore` objects store the scores for each student. You are advised to decompose this method into simpler parts and implement one or more `private` helper methods to keep the code more understandable.
- A getter method for the `Assignment` field.

You may write a stand-alone `TestAssignmentLoader` class, or you can proceed to the next part of the assignment where all previous parts of the assignment will be brought together to make a final application. (Or rather: a simple, limited-functionality application that demonstrates how these current classes would be tied together if we were to continue to build a proper application).