# PBL: Grade Book Part 4: `class StudentLoader`



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

You should have completed coding and testing `class Student`, `class Assignment`, and `class StudentFileWriter`.

For this part of the project, we will study write that will read the student file containing the list of students for the grade book – the file that was generated by `class StudentFileWriter`.

Study and understand the following class that will test `class StudentLoader`, as well as output that was obtained from running it.

```
import java.util.ArrayList;
public class TestStudentLoader {
    public static void main(String[] args) {
        System.out.println("class TestStudentLoader");
        ArrayList<Student> list = StudentLoader.load("students.txt");
        for(Student s : list) {
            System.out.println(s);
        }
    }
}
```

```
class TestStudentLoader
File read complete.
1234 Wong, Cynthia
2345 Huang, James J.
3456 Long, Chen
```

The test class is testing the StudentLoader class. Answer these questions before reading further:

- Does the test class instantiate an object of type `StudentLoader`? Worded in another way: is an instance of `StudentLoader` class created? Or: is a `StudentLoader` object created?
- What methods from the `StudentLoader` class are tested?
- For each `StudentLoader` method tested, is an instance of the class used to call the method, and if not, how is the method called?
- For each `StudentLoader` method tested, what parameters are passed to the method?
- For each `StudentLoader` method tested, what data type is returned from the method?
- For each `StudentLoader` method tested, what does the test code do with the data returned from the method?
- For the output shown, what output is printed by the `TestStudentLoader` class?
- For the output shown, what output is not printed by the `TestStudentLoader` class, and where does that output from?

Continue reading only after you have understood and answered all the above questions.

The first thing you should notice is that this test class is not instantiating any object of type `StudentLoader`. The test method is calling a single method named `load` from the `StudentLoader` class. This `load` method is called using the `StudentLoader` class name to inform the compiler where that `load` method is to be found. If a method can be called using the class name rather than an instance of the class, the method is necessarily a `static` method.

## PBL: Grade Book Part 4: `class StudentLoader`



The `load` method takes as a parameter a `String` object, apparently the name of the file to be loaded, returns an `ArrayList` that contains `Student` objects. You should now be able to write the method header for this file. Try to do this.

Since the method will return an `ArrayList` of `Student` objects, it should be obvious that we will need to declare and instantiate an `ArrayList` object that will be returned after populating it with students. Try to write the line of code now.

Much of the remainder of the method load is given on the following page.

The method declares a variable named `scanner` of type `Scanner` and initializes it to `null`.

An attempt to execute the code within the `try` block will be made. If any errors occur within this block, execution of the code will cease, and execution will jump to execute the code inside the `catch` block. The code within the `catch` block will only be executed if the code within the `try` block has an error. After the code in either the `try` or `catch` block has completed, the code of the `finally` block is executed. In this block, the `Scanner` object is closed. We need to close the file whether or not there has been an error in reading the file.

The code for the loop to call a `scanStudent` method has not been implemented. The `scanStudent` method can be made `private` because it is a helper method intended to only be called by the `load` method from within the `StudentLoader` class. The `scanStudent` method will need to take the file `Scanner` object as a parameter, use that scanner to read and parse the fields to make a Student object, then return that student object. The format of the file is to be the format that was written by `class StudentFileWriter`. Write the loop inside the `load` method and the `scanStudent` method.

# PBL: Grade Book Part 4: `class StudentLoader`



```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Scanner;

public class StudentLoader {

    // Method header for: load
    
        // Declare list, an ArrayList to store Student objects
        
        Scanner scanner = null;

        // The try block will catch errors when parsing file
        try {

            scanner = new Scanner(new File(filename));
            scanner.useDelimiter("[,\\n]"); // For CSV

            // Build the list of students by repeatedly calling
            // method scanStudent until null is returned.
            

        } catch (FileNotFoundException e) {
            System.out.println("File not found!");
        } catch (Exception e) {
        System.out.println(
            "An error occurred while reading the file: " +
            e.getMessage());
        } finally {
            if (scanner != null) {
                scanner.close();
            }
        }

        return list;

    }
```