AP Computer Science A – Writing Classes English Name: ____________________

PBL: Grade Book Part 5: `class StudentList`

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`, `class StudentFileWriter`, and `class StudentList`.

The class `StudentList` will be a wrapper class that implements methods relating to the list of `Student` object. The class will require `class Student` and `class StudentLoader`.

For this part of the project, we will write a wrapper class to contain the list of students and provide methods that will perform operations we may need to perform on that list, such as getting the `Student` object that is associated with a given student number.

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

```
import java.util.Arrays;

public class TestStudentList {

    public static void main(String[] args) {

        StudentList studentList = new StudentList();
        studentList.load("apcsc_students.txt");

        Student[] studentArray = studentList.getStudentArray();
        System.out.println(Arrays.toString(studentArray) + "\n");

        System.out.print(studentList);

        Student s = studentList.getStudent(2345);
        System.out.println("\nStudent found:\n" + s);

    }

}
```

```
[1234 Wong, Cynthia, 2345 Huang, James J., 3456 Long, Chen]

1234 Wong, Cynthia
2345 Huang, James J.
3456 Long, Chen

Student found:
2345 Huang, James J.
```

Answer these questions before reading further:

1. Is an instance of `StudentList` created?
   - Yes, and instance of `StudentList` is created.
2. What parameters does the `StudentList` constructor take?
   - The constructor does not take any parameters
3. What `StudentList` methods does the test code invoke? For each:
   - Is the method an instance method or a static method?
   - What parameters does the method take?
   - Does the method produce any output to the console?
   - What is the return type of the method?
   - `load`
     - instance method
     - takes a String parameters
     - does not produce any output to the console
     - there is no return value from the method.
   - `getStudentArray`
     - instance method
     - does not take any parameters
     - does not produce any output to the console
     - returns an array of `Student` objects
   - `toString`
     - instance method
     - does not take any parameters
     - prints a list of the students
     - returns a `String` object
   - `getStudent`
     - instance method
     - takes an integer – the student number – as a parameter
     - does not produce any output to the console
     - returns a `Student` object

After studying the test code, you are ready to write `class StudentList`. Here are additional specifications and hints:

- You will need to import `java.util.ArrayList`.
- The class will need an instance field to store the list of students after it has been loaded.
- The `load` method in the `StudentList` class can use the `StudentLoader.load` method to load the student list from a file.
- For the `getStudentArray` method, the `ArrayList` class has a `toArray` method that will return an array of the element object. In order to return an array of `Student`, call the method in the following way:

  ```
  Student[] studentArray = studentList.toArray(new Student[0]);
  ```
- For the `getStudent` method, a simple linear search through the array using a `for` loop is likely the most prudent way to find the student element.
- For the `toString` method in the `StudentList` class, you can call the `toString` method from the `Student` class to obtain the `String` object for each student, and append each of these to generate the `String` object for the entire student list.

```
import java.util.ArrayList;

public class StudentList {

    private ArrayList<Student> studentList;

    public StudentList() {
        this.studentList = new ArrayList<>();
    }

    public int load(String filename) {
        studentList = StudentLoader.load(filename);
        return studentList.size();
    }

    public Student[] getStudentArray() {
        return studentList.toArray(new Student[0]);
    }

    public void print() {
        for(Student s : studentList) {
            System.out.println(s);
        }
    }

    public Student getStudent(int studentNumber) {
        for(Student s : studentList) {
            if(s.getStudentNumber() == studentNumber) {
                return s;
            }
        }
        return null;
    }

    public String toString() {
        String str = "";
        for(Student s : studentList) {
            str += s.toString() + "\n";
        }
        return str;
    }

}
```