Unit 05 Quiz – FRQ SOLUTION

Points will be given for neatness, indentation, and proper naming conventions. Variables and methods should have descriptive naming.

Consider the following description of the `Student` class which includes four fields, two constructors, accessor and mutator methods, and a `toString` method. All fields must be `private` unless otherwise specified. All methods must be accessible from outside the class, unless otherwise specified. The class must contain:

- At least one `static` field, `numStudents` that represents the total number of `Student` objects that were created (this is accomplished by initializing the value to zero and incrementing the value in the constructors).
- At least three instance fields: `name`, a string representing the student's full name, `birthYear`, an integer representing the year the student was born, and `gpa`, a number of type `double` that represents the GPA of the student.
- A three-parameter constructor that takes in and sets initial values for `name`, `birthYear`, and `gpa`.
- A two-parameter constructor that takes in and sets initial values for `name` and `birthYear`.
- A method `getNumStudents` that returns the value contained in `numStudents`. This method must be able to be called using the class name (i.e.: without the need to create an instance of the class).
- Accessor methods for `name`, `birthYear` and `gpa`.
- A mutator method for `gpa`.
- A `toString` method that can be used to get a string representation of the object (so that the information contained in the object can be printed using `System.out.print`). The output should conform to the output shown for the test code. Take note of the quotation marks around the name, and where newline characters are placed. Also, as shown in the example output, if the value of the `gpa` field is `0.0` for a student (i.e.: the GPA has not been set), their GPA should not be included in the string representation.

The following is test code that must work correctly with your code. The output of your code should conform to the specification above, and when run with the test code below, must produce the output shown.

```
// Test static variable
System.out.print("Number of Students: ");
System.out.println(Student.getNumStudents() + "\n");

// Test three-parameter constructor
Student anna = new Student("Anna Cao", 2006, 3.8);
System.out.println(anna);

// Test static variable
System.out.print("Number of Students: ");
System.out.println(Student.getNumStudents() + "\n");

// Test two-parameter constructor
Student cindy = new Student("Cindy Zhang", 2007);
System.out.println(cindy);

// Test static variable
System.out.print("Number of Students: ");
System.out.println(Student.getNumStudents() + "\n");

// Update attributes and retest
cindy.setGpa(3.7);
System.out.println(cindy);

// Test static variable
System.out.print("Number of Students: ");
System.out.println(Student.getNumStudents() + "\n");
```

Unit 05 Quiz – FRQ SOLUTION

Output of the test code:

```
Number of Students: 0

Name: "Anna Cao"
Birth year: 2006
GPA: 3.8

Number of Students: 1

Name: "Cindy Zhang"
Birth year: 2007

Number of Students: 2

Name: "Cindy Zhang"
Birth year: 2007
GPA: 3.7

Number of Students: 2
```

Write your code in the space below and on the following pages.

```
public class Student {

    // Class Fields (Class Attributes)
    private static int numStudents = 0;

    // Instance Fields (Instance Attributes)
    private String name;
    private int birthYear;
    private double gpa;

    // Three-parameter Constructor
    public Student(String name, int birthYear, double gpa) {
        this.name = name;
        this.birthYear = birthYear;
        this.gpa = gpa;
        numStudents++;
    }

    // Two-parameter Constructor
    public Student(String name, int birthYear) {
        this.name = name;
        this.birthYear = birthYear;
        this.gpa = 0.0;
        numStudents++;
    }
```

Unit 05 Quiz – FRQ SOLUTION


```
    public static int getNumStudents() {
        return numStudents;
    }

    // Getter and Setter methods
    public String getName() {
        return name;
    }

    public int getBirthYear() {
        return birthYear;
    }

    public double getGpa() {
        return gpa;
    }

    public void setGpa(double gpa) {
        this.gpa = gpa;
    }

    // toString method
    @Override
    public String toString() {
       String s = "Name: \"" + name + "\"\n" +
                  "Birth year: " + birthYear + "\n";
       if(gpa > 0.0) {
          s += "GPA: " + gpa + "\n";
       }
       return s;
    }

}
```