Instructions: **Class GPA** – You are to write a program that keeps track of the GPAs of the students in a course. Each student has a name, and a GPA (between 0 and 4.0).

You are to write a program that asks the user to enter a student's name and their GPA, and then create a Student object for each student. Your student object should be created using a constructor to initialize all values.

Additionally, your Student class should keep track of the overall class GPA, as well as the highest GPA and the name of the student with said GPA.

The user should continue to enter students information until they enter a blank string for the student's name (sentinel value).

Once the user has finished entering students information, the program should print a report with the amount of students in the class, the average GPA, the highest GPA, and the student with the highest GPA.

You program should contain 2 classes, **Student** and **Course**, defined as follows:

**Student:**

- This class will have two instance variables:

  ```java
  String name
  double gpa
  ```

- and four static variables:

  ```java
  int numberOfStudents
  double averageGpa
  String highestGpaName
  double highestGpa
  ```

- You must not add any additional variables to the class.

- The class will have 1 method:

  ```java
  public static void printReport() - This will print a report containing the total number of students, the average GPA, the highest GPA, and the name of the student that had the highest GPA.
  ```

- The class will have a constructor to take the student's name and GPA.
Course:

- This class will only have a main method. It should repeatedly prompt the user to enter information for a student, and create a student object for each. Once the user has completed entering information, the program will print a report before exiting.

Demonstrate your game to the TA to receive your grade.

Considerations (please read):

When keeping track of a minimal or maximal value, it is good practice to continually store that number in a single variable and to first initialize that variable with the maximal or minimal value (respectively) that your system can represent.

For example, if a variable (int max) is keeping track of the largest integer value seen so far from a stream of numbers, max should first be initialized with a very low number so as to ensure that the very first number that max is compared to will be higher than the number used to initialize it. The converse is true when keeping track of the smallest number that's been seen.

In this example, a good number to use to initialize our max variable is Integer.MIN_VALUE. This static constant found in the Integer class holds the smallest possible integer value in our program (conversely, Integer.MAX_VALUE stores the largest possible integer value). Therefore, when we initialize our variable max to Integer.MIN_VALUE, the first time we compare a number to max, for instance:

\[
\text{if (current\_val} \geq \text{max) max = current\_val;}
\]

it is guaranteed to be higher than or equal to the value of max and therefore the max variable will be updated.

If max were initialized to another value, such as zero, then we could easily end up with a logical error if, for example, current_val was -1, because -1 would have been the largest value we have seen so far, but since it is not greater than zero, our program would not store in the the max variable.