Q1

Assume that you are part of a team working on an app that fixes errors in written text.

Your job is to write a Java program that prompts the user to input some text, then cleans up the text according to the following rules before writing it to the console:

All sentences start with an uppercase letter.

If the sentence contains the word Java it should be uppercase e.g. JAVA

Other than the word Java and the first letter in a sentence, all remaining letters are lowercase.

There should be one space between words.

There should be no spaces before a full stop or a comma.

There should be one space after a full stop or comma (unless it is the final character of the text as per rule 7 below).

There should be no spaces at the beginning and end of the text.

For example, if the user entered the following text:

“ i love java. It IS THE best programming Language , and NO other language is as good. “

Your program would output:

“I love JAVA. It is the best programming language, and no other language is as good.“

Please note the double quotes in the text above are to show you the beginning and end of the text and are not part of the example text.

Your program must also output the number of characters that were changed. Please note that the following counts as changing a character:

Removing a space

Adding a space

Changing a character to uppercase

Changing a character to lowercase

Other requirements

Make sure you:

Add class comments, with your name as author and date as the last date you worked on this exercise.

Add comments to describe your code.

Name all variables and methods as per Java naming conventions.

Q2

Assume you need to develop a program for the university to keep track of:

* Students in a unit and
* Units in a degree.

As the programmer your job is to:

**Complete the Student class**

The Student class stores details about a student. The Student class needs:

* At least three instance variables that are suitable for a student
* Constructor(s)
* Getters and Setters

**Complete the Unit class**

The Unit class needs to store (i.e. instance variables):

* The unit name
* The students in the unit (An ArrayList of Student objects i.e. of data type Student)

The Unit class also needs to be able to create a Unit i.e. a constructor(s).

Methods are needed to:

* Get and set the unit name
* Add a student to the unit (the ArrayList)
* Remove a student from the unit (the ArrayList). The method should print a message confirming if the student was removed or not removed.
* Check if there is a particular student in the unit (the ArrayList). The method should print a message confirming if the student is in the unit or not in the unit.
* Return the number of students in the unit (the ArrayList)
* Remove all students from the unit (the ArrayList). The method should print a message confirming if all students were removed or not removed.
* Print all students in the unit (the ArrayList)

**Complete the Degree class**

The degree class needs to store (i.e. instance variables):

* The degree name
* A collection of key value pairs (i.e., a HashMap) where the key is the unit name (i.e. a string) and the value is a Unit object i.e. of data type Unit.

The Degree class also needs to be able to create a Degree i.e. a constructor(s).

Methods are needed to:

* Get and set the degree name
* Add a unit to the degree (the HashMap)
* Remove a unit from the degree (the HashMap). The method should print a message confirming if the unit was removed or not removed.
* Check if there is a particular unit in the degree (the HashMap). The method should print a message confirming if the unit is in the degree or not in the degree.
* Return the number of units in the degree (the HashMap)
* Remove all units from the degree (the HashMap). The method should print a message confirming if all units were removed or not removed.
* Print all units in the degree (the HashMap)

**Complete the QuestionTwo class**

In the main method:

* Create at least one Student
* Create at least one Unit and add the Student to the Unit
* Create at least one Degree and add the Unit to the Degree

**Other requirements**

Make sure you:

* Add class comments, with your name as author and date as the last date you worked on this exercise.
* Add comments to describe your code.
* Name all variables and methods as per Java naming conventions.

Q3

Assume that you need to write an application for a computer shop. The application requires a generic Computer class and other classes that extend the Computer class and are designed for specific types of computer. You need to:

**Complete the Computer class** that stores details about a computer. The Computer class needs:

* At least three instance variables that are suitable for a computer
* Multiple constructors
* Getters and Setters
* A method that prints the computers details. The output must be formatted for readability.
* Another method that is appropriate for a generic computer class

**Complete the Laptop class**. You must extend the Computer class. In the Laptop class write the code for:

* A minimum of 3 instance variables that are appropriate for a laptop
* Multiple constructors
* Getters and Setters
* A method that prints the laptop details. The output must be formatted for readability.
* Any other methods you think are required to meet the requirements below

In the Laptop class you must demonstrate the following:

* The use of the super keyword in a constructor AND a method
* Overloaded methods
* Overriding a method

**Complete the QuestionThree class**. In the main method:

* Create at least one Laptop object
* Create at least one Computer object
* Use your overloaded methods in the Laptop class
* Use the methods in your Computer class and Laptop class that demonstrate method overriding

**Other requirements**

Make sure you:

* Add class comments, with your name as author and date as the last date you worked on this exercise.
* Add comments to describe your code.
* Name all variables and methods as per Java naming conventions.