In our lecture, we discussed a Vector class, and came up with the following code.
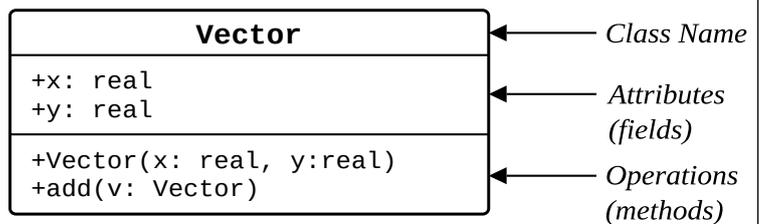
Code Block 1*: The* Vector *class*

```
1  public class Vector {
2      public double x;
3      public double y;
4      public Vector(double x, double y) {
5          this.x = x;
6          this.y = y;
7      }
8      public void add(Vector v) {
9          this.x += v.x;
10         this.y += v.y;
11     }
12 }
```

The UML class diagram for the code above is given here, immediately to the right. Recall the plus symbol (+) denotes the field or method is public, and, in Java, we use the primitive type double to store real number values. Study this diagram well and compare it to the code above.

UML class diagram for the Vector class:

Vector	← Class Name
+x: real +y: real	← Attributes (fields)
+Vector(x: real, y:real) +add(v: Vector)	← Operations (methods)

The Java statement to the right instantiates an object of type Vector, and assign a reference to that object to a variable with identifier a. The diagram below the statement represents the data structure that is created by that statement.

Code to instantiate a Vector object, and a diagram representing the resulting object in memory:

```
Vector a = new Vector(1.5, 2.1);
```

Vector a		Vector	
●→		double x	1.5
		double y	2.1
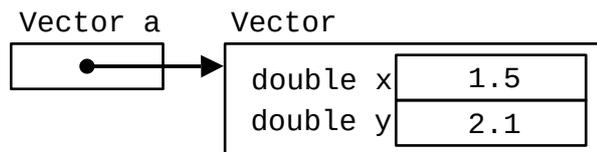
Please refer back to the code and diagrams above to assist you in answering the questions in this assignment.

1. A student has created a Book class. The class contains a field to represent an integer variable called pages that is to represent the number of pages, and a boolean field called isHardcover that is to indicate whether or not the book is hardcover. An object with the identifier story will be declared as type Book. Assume all fields, methods, and the class itself are public.

    a) Complete the UML class diagram to the right according to the specification given in the question.
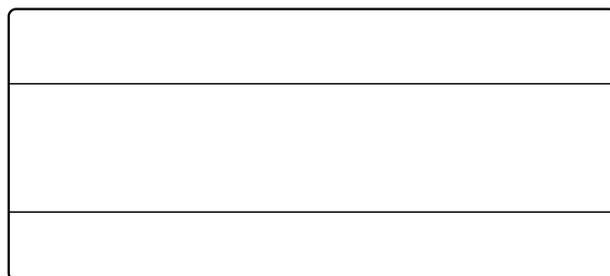
b) Label the diagram below representing the object `story` of type `Book`.
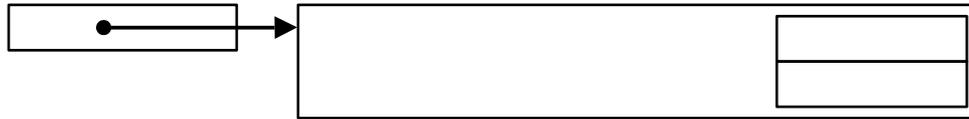
c) In the box to the right, write Java code for the `Book` class, as specified in the question. Assume there is no explicit constructor written for the class. Pay attention to details, including appropriately terminating statements with semicolons (`;`) and proper indentation.

d) Write the Java statement that will declare an object `story` of type `Book` by calling the default constructor.

2. A teacher has created a `Student` class. The class contains the following:
   - A private `int` field called `grade` to represent the student's grade level
   - A private `String` field called `name` to represent the student's name
   - A private `double` variable called `average` to represent the student's grade-point average
   - A public method called `updateAverage` that does not return any value, but takes a parameter of type `double` and updates the student's average.
   - A public constructor that takes an `int` parameter called `grade` and a `String` parameter called `name`.

   a) Complete the UML class diagram according to the specification given in the question. Remember that private fields and methods are preceded by a minus symbol (`-`).

b) Label the diagram below representing an object `greg` of type `Student`.

9
Greg
90.5
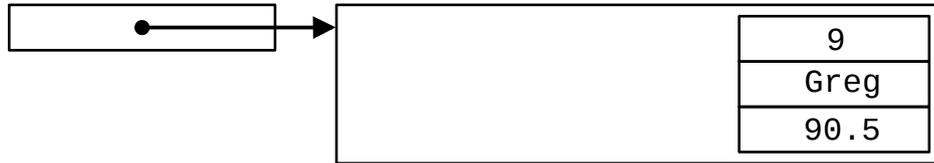
c) In the box to below, write Java code for the `Student` class. Include a constructor that will set the relevant fields based on its parameter values, and a method `updateAverage` that will set the `average` field based on its parameter value.

d) Write a Java statement that will declare an object `greg` of type `Student`.

e) Write a Java statement that calls `updateAverage` to update the average of `greg` to the value of `93.4`. Remembering how `String` class methods are called may help.