Objects Containing Objects 2

Create a Java program that imitates a schoolbag containing two binders. The program should contain three classes: `Binder`, `SchoolBag` and `BagTester` (which consists of the main method).

1. The `Binder` class.
    a. Create all the necessary instance fields for the class. Each `Binder` object should keep track the following information:
        - Label
        - Color
        - Number of sheets of paper in the binder
    b. Create the class constants for the class:
        - Weight of each sheet of paper
        - Weight of an empty binder
    c. Create all the necessary methods. Each `Binder` object should perform the following tasks:
        - Add sheets to the binder
        - Calculate the total weight of the binder (with the sheets in it)
        - Comparing the weight of the implicit and explicit binder
        - Allows the capability to print info about the binder
    d. Create a constructor that initializes the instance variables of the object being constructed.
    e. Create the toString that return the String representation of the Binder, including all values of all fields with descriptions, e.g.,

```
Label: English
Color: Red
# of sheets: 21
```

2. The `SchoolBag` class:
    a. A schoolbag has a style (String) and consists of two binders. Create the instance fields accordingly.
    b. Create the class constants for the class:
        - Weight of the empty schoolbag
    c. Create the methods that perform the following tasks:
        - Add sheet(s) to given binder (specified by id / index)
        - Remove sheet(s) from given binder
        - Calculate the average weight of a binder in the bag
        - Calculate the total weight of the schoolbag (with the binders)
        - Allows the capability to print info about the schoolbag
    d. Create two constructors:
        - The first one takes two object reference parameters. The two objects will be the two binders. The object must already exist before using this constructor.

- The second one takes as parameters all the information of the two binders. The two binders will be created in the constructor and assigned to the corresponding fields.

e. Create the toString method that returns the string representation of the Bag, including values of all fields and descriptions.

3. Create a `BagTester` class that performs the following:
   a. Prompt user for information of two school bags and their content
   b. Create two instances of school bag. Use different constructors to create each instance
   c. Continue to prompt user for the following until -1 is entered for ID
      - ID for a schoolbag
      - ID for a binder in the schoolbag
      - Add or remove sheets
      - The number of sheets

Sample Output:

```
Please select a schoolbag (by ID, 0 or 1): 0
Please select a binder in schoolbag 0 (by ID, 0 or 1): 1
Add (a) or remove (r) sheets: a
# of sheets: 3

Please select a schoolbag (by ID, 0 or 1): 1
Please select a binder in schoolbag 1 (by ID, 0 or 1): 1
Add (a) or remove (r) sheets: r
# of sheets: 1

Please select a schoolbag (by ID, 0 or 1): 0
Please select a binder in schoolbag 0 (by ID, 0 or 1): 0
Add (a) or remove (r) sheets: r
# of sheets: 2

Please select a schoolbag (by ID, 0 or 1): -1
```

   d. Print out the information of each schoolbag

Sample Output:

```
SchoolBag #0
Style: Nike blue
Binder #0 -
     Label: Math
     Color: Red
     # of sheets: 0
Binder #1 -
```

Label: English
Color: Blue
# of sheets: 10