CS 61B Spring 2023 — Introduction to Java

Exam-Level 2: January 23, 2023

1 On Wednesdays We Wear Pink

Sherry is trying to create `Closet` class to organize her all her clothes! Given the `Clothing` and `Closet` classes below, complete the following tasks using the appropriate data structures, variables, and functions.

```
class Clothing {
    public String type; // one of "top", "bottom", "shoes", or "accessory"
    public String color; // describes the color of the item

    public Clothing(String type, String color) {
        this.type = type;
        this.color = color;
    }

    public void dyeColor(String newColor) {
        ...
    }
}

class Closet {
    ...
}
```

1. Sherry's closet is too messy, and she's forgotten which items of clothing she owns! Luckily, she gives you a log of all the clothes she's worn recently as a `List<Clothing>`. For example, given the `List` {`"Green Top", "Yellow Shoes", "Yellow Shoes"`}, the final data structure should only include {`"Green Top", "Yellow Shoes"`} once each.

   Given this `List`, write a constructor and add necessary instance variable(s) to populate the `Closet` with the **unique** items she owns.

2. Sherry is all about color coordination–after all, everyone knows you can't wear yellow and purple together! Using what you have from part (1), add an instance variable that allows you to quickly retrieve all items of the same color. Also, add code to the constructor as necessary to populate this new instance variable.

3. Being very fashionable, Sherry only wears specific colors on specific days. She decides to add an `getItemsByDay` method to the `Closet` class. It takes in a `Map<String, String>` which has keys as days of the week and values as colors, as well as the current day of the week, and returns the valid items of clothing. For example, given a mapping {`"Wednesday": "Pink", "Friday": "Maroon",`

`"Tuesday": "Lavender"}` and a `currentDay` of `"Wednesday"`, `getItemsByDay` should return all the pink items in the `Closet`.

Using the instance variables you have created, fill in the `getItemsByDay` method.

```
class Closet {
    ... // constructor and instance variables

    public List<Clothing> getItemsByDay(Map<String, String> daysToColors, String currentDay) {
        ... // fill this in!
    }
}

```

4. Sherry just got her heart broken and is entering her emo phase. Given a `List<String> happyColors`, dye any clothes with a "happy color" to `"Black"`. For example, given a `List {"Red", "Pink"}`, any clothes that are `"Red"` or `"Pink"` should have their color changed to `"Black"`. Remember to use methods of the `Clothing` class to do this!

```
class Closet {
    ... // previous code

    public void enterEmoPhase(List<String> happyColors) {
        ... // fill this in!
    }
}

```

2 Static Books

Suppose we have the following `Book` and `Library` classes.

```
class Book {
    public String title;
    public Library library;
    public static Book last = null;

    public Book(String name) {
        title = name;
        last = this;
        library = null;
    }

    public static String lastBookTitle() {
        return last.title;
    }
    public String getTitle() {
        return title;
    }
}
```

```
class Library {
    public Book[] books;
    public int index;
    public static int totalBooks = 0;

    public Library(int size) {
        books = new Book[size];
        index = 0;
    }

    public void addBook(Book book) {
        books[index] = book;
        index++;
        totalBooks++;
        book.library = this;
    }
}
```

(a) For each modification below, determine whether the code of the `Library` and `Book` classes will compile or error if we **only** made that modification, i.e. treat each modification independently.

1. Change the `totalBooks` variable to **non static**
2. Change the `lastBookTitle` method to **non static**
3. Change the `addBook` method to **static**
4. Change the `last` variable to **non static**
5. Change the `library` variable to **static**

(b) Using the `Book` and `Library` classes from before, write the output of the `main` method below. If a line errors, put the precise reason it errors and continue execution.

```
1  public class Main {
2      public static void main(String[] args) {
3          System.out.println(Library.totalBooks);                  _____________________
4          System.out.println(Book.lastBookTitle());                _____________________
5          System.out.println(Book.getTitle());                     _____________________
6
7          Book goneGirl = new Book("Gone Girl");
8          Book fightClub = new Book("Fight Club");
9
10         System.out.println(goneGirl.title);                      _____________________
11         System.out.println(Book.lastBookTitle());                _____________________
12         System.out.println(fightClub.lastBookTitle());           _____________________
13         System.out.println(goneGirl.last.title);                 _____________________
14
15         Library libraryA = new Library(1);
16         Library libraryB = new Library(2);
17         libraryA.addBook(goneGirl);
18
19         System.out.println(libraryA.index);                      _____________________
20         System.out.println(libraryA.totalBooks);                 _____________________
21
22         libraryA.totalBooks = 0;
23         libraryB.addBook(fightClub);
24         libraryB.addBook(goneGirl);
25
26         System.out.println(libraryB.index);                      _____________________
27         System.out.println(Library.totalBooks);                  _____________________
28         System.out.println(goneGirl.library.books[0].title);     _____________________
29     }
30 }
```