COS 126 General Computer Science Spring 2012

Programming Exam 2

This test has 1 question. You have 50 minutes. The exam is open book, open note, and open web. You may use code from your programming assignments or the Introduction to Programming in Java booksite. No communication with any non-staff members is permitted. Submit your solution via Dropbox. **Write out and sign the Honor Code pledge before turning in the test.**

"I pledge my honor that I have not violated the Honor Code during this examination."

```
------------------------------
Signature
```

Name:

NetID:

Precept:

Do not remove this exam from the exam room.

Your task. Your task is to implement a ”self-organizing” list. We will call it a PList (for Popularity List). It is simply a list of entries, where each is an object from the `Entry` class described at the bottom of this page, but when you look something up (`find`) or update an existing entry (`update`), that entry goes to the top of the list. New things (`add`) always go to the top.

You may write your `PList` implementation using your own linked list code or using `Stack.java` from the Booksite. (Hint: you will need two stacks.)

http://introcs.cs.princeton.edu/java/43stack/Stack.java

For a penalty of 5 points, you may implement it as an array of `Entry` objects with a fixed array size of 200.

If you go with the linked list option, you will need a private inner `Node` class defined as follows:

```
private class Node {
    private Entry en;
    private Node next;
}
```

API Specification

```
public class PList
---------------------------------------------------------------------------------------------------
        PList()                            // construct an empty Popularity List

 void   add(String name, String num)       // make an Entry for name and number
                                           // add that Entry to the top of PList
                                           // no need to check if entry is already on PList

 void   show()                             // output PList names and numbers
                                           // in the list order (most recently accessed first)
                                           // output the message "empty PList" if it is empty

 String find(String name)                  // move Entry with name to top of PList
                                           // return number
                                           // if Entry not there, return null

 Entry  cut(String name)                   // remove and return Entry from the PList
                                           // if not there, throw new RuntimeException()

 void   update(String name, String num)   // if Entry is not on the list, add it
                                           // otherwise, remove the old Entry and
                                           // add a new Entry at the top of the list
```

Helper class. You are given a class `Entry` that implements the following API:

```
public class Entry
---------------------------------------------------------------------------------------------------
        Entry(String name, String num) // construct an Entry using the arguments
String  getName()                      // return name
String  getNum()                       // return num
boolean equals(String name)            // Is this Entry's name equal to name?
String  toString()                     // return String representation of Entry
```

You may copy or download it from

`http://www.cs.princeton.edu/courses/archive/spring12/cos126/docs/Entry.java`

Testing. Here is a test `main()` that tests each of your methods in `PList`.

You may copy or download it from

`http://www.cs.princeton.edu/courses/archive/spring12/cos126/docs/PList.main.txt`

```
   // test main
   public static void main(String[] args) {
       PList list = new PList();
       list.add("Doug", "6092586314");
       list.add("Donna", "6092581978");
       list.add("Christopher", "6092585388");
       list.add("Maia", "6092586484");
       list.add("Chris", "6092582038");
       StdOut.println("Entire list:");
       list.show();
       StdOut.println("===========================");

       list.update("Maia", "6092588888");
       list.cut("Chris");
       StdOut.println("Doug " + list.find("Doug"));
       StdOut.println("Maria " + list.find("Maria"));
       StdOut.println("===========================");

       StdOut.println("Updated list:");
       list.show();
   }
```

Sample Run. Here is our output for the test `main()` given above:

```
> java PList
Entire list:
Chris 6092582038
Maia 6092586484
Christopher 6092585388
Donna 6092581978
Doug 6092586314
===========================
Doug 6092586314
Maria null
===========================
Updated list:
Doug 6092586314
Maia 6092588888
Christopher 6092585388
Donna 6092581978
```

Your program should work for other clients as well.

Submission. Submit the file `PList.java` via `Dropbox` at

```
https://dropbox.cs.princeton.edu/COS126_S2012/Exam2
```

Be sure to click the *Check All Submitted Files* button to verify your submission.

Grading and Advice. *Your program will be graded on correctness and clarity. You will receive partial credit for correctly implementing the following components:*

- *Header with name, login, precept*: 1 point.
- *Well-organized and commented code*: 1 point.
- *Instance variables, constructor,* `add()`*, and* `show()`: 15 points.
- `cut()`, `update()`*, and* `find()`: 13 points.

You get points for each method that has the correct signature and return statement. You get additional points for completing each method.

`find()` *is the most involved of the methods, so be sure to budget your time appropriately.*

You should not need to retype large blocks of code.

You will receive a substantial penalty if your program does not compile or if you do not follow the prescribed API or output specification.