CS145 – PROGRAMMING ASSIGNMENT
CARD HEAP/PRIORITY QUEUE GUI

MORE WITH CARDS

OVERVIEW

This program focuses on heaps and using the GUI.

INSTRUCTIONS

Turn in `Card.java`, `CardHeap.java` and your main file. You are allowed to reuse your `Card.java` from a prior assignment with any necessary changes that you want to add.

3 Files should be submitted!

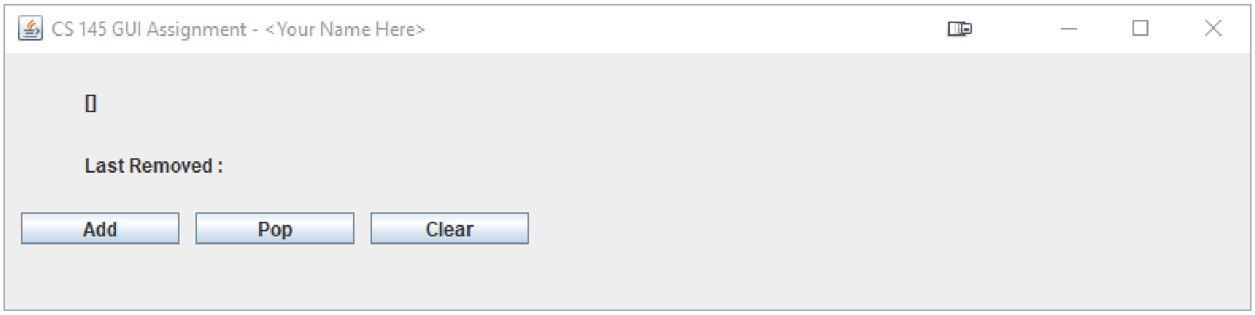
IMPLEMENTATION DETAILS:

You will reuse your `Card.java` file and you will write the GUI form and the `CardHeap.java` files to implement the necessary behavior.

THE PROGRAM

In this assignment, you will implement a simple GUI program that will randomly add/remove cards that are the same as the ones in prior assignments. *(Note that we will be ignoring the premium cards, although they should still work.).*

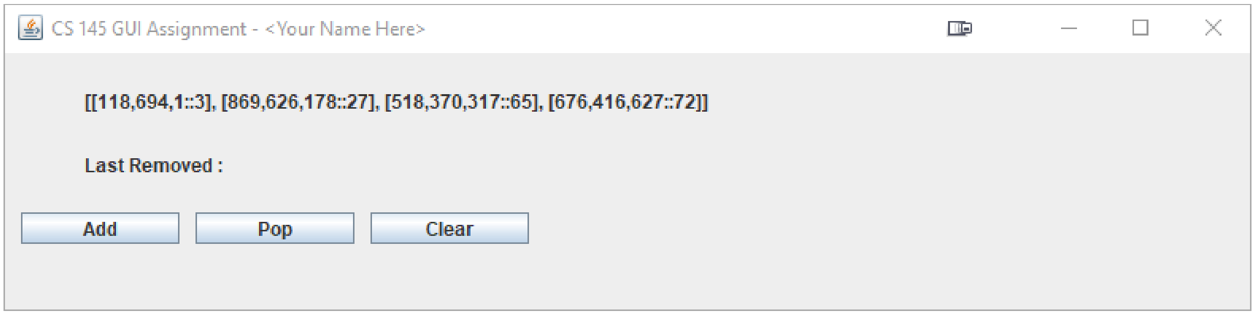
Your program should look like the following at the beginning:



```
[]

Last Removed :
```

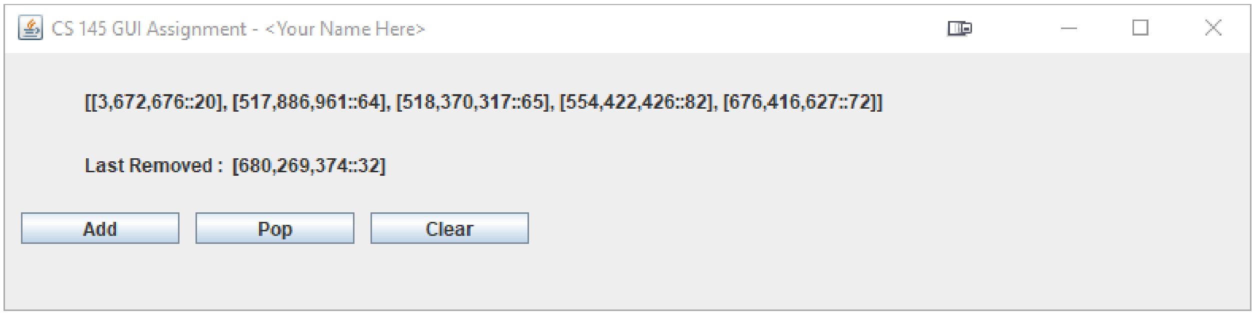
After a few adds:



```
[[118,694,1::3], [869,626,178::27], [518,370,317::65], [676,416,627::72]]

Last Removed :
```

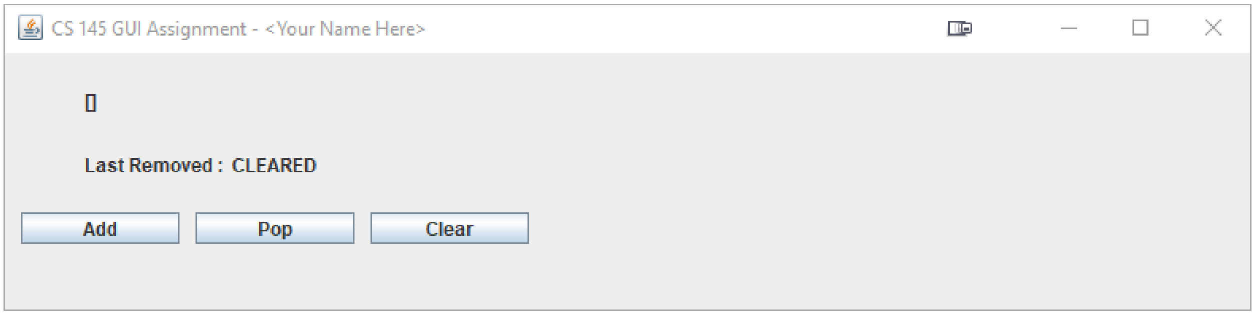
After a few rounds of adding and removal:



```
[[3,672,676::20], [517,886,961::64], [518,370,317::65], [554,422,426::82], [676,416,627::72]]

Last Removed :  [680,269,374::32]
```

After a clear:



```
[]

Last Removed :  CLEARED
```

THE MAIN PROGRAM

The GUI/main program should do the following:

- The add button should create a random card*(default constructor)* and then add it to the heap.
- Pop should remove the smallest card from the heap.
  - It should then print the removed card in the label marked Last Removed
- Clear should set the heap to empty.
- Every time you press a button it should reset the top label to the toString() representation of the storage heap. In other words keep the labels updated.
- When the program starts, the 2nd label can be blank if you want.

CARD

You should not need to make any significant change to the Card class, just make sure the compareTo method works correctly and the toString works correctly.

Note you should be using compareTo in you heap, that is the cleanest solution.

THE HEAP/PRIORITY QUEUE

The heap should be a stored in an Array of Cards. I would **not** suggest trying to make a generic heap. You can use the PowerPoint/book as a guide on how to write it.

The required methods you will need are:

- `public void add(Card x)`
- `public Card remove()`
- `public void clear()`
- `private void expand()`
- `public String toString()`
  - Only print the used part of the array, do not print the entire storage array.