Assignment Prefix: Lab02

Due Date: Friday, Jan. 27th, 2017 @ 11:59pm

Points: 100

This is an individual assignment.

Create a NetBeans project named Lab02 and ensure it is saved to a location like desktop or your flash drive.

Caution:

The cluster computers are reimaged every morning between 3:00am and 5:00am. Any files left on a cluster computer will not be there the next day.

Blackboard has a file tab that provides a limited amount of cloud storage.

Restrictions:

Unless directed otherwise, you cannot use any Java class libraries in this assignment. In particular, you cannot use the ArrayList class nor can you use any methods from the Arrays class.

If needed, you may create and use one or more instances of an array and access the length instance variable for each array.

In this project you will be doing the following:

Develop a ***java interface*** named ***Bag*** that can store a certain number of whole numbers in it. A ***bag*** in java is a kind of ***collection*** that does not do much more than contain its items. It does not order the items in any particular order and nor does it prevent duplicates.
Provide the following methods in the interface.

- ***getCurrentSize( )*** that returns a count of numbers in the ***bag***
- ***isEmpty( )*** – that checks if the ***bag*** is empty, returns true when empty
- ***add (int num)*** – adds a new number ***num*** to the ***bag***
- ***remove (int num)*** – removes the first occurrence of the number ***num*** from the bag
- ***remove( )*** – removes a randomly selected entry from the bag
- ***clear( )*** – removes all the numbers from the bag
- ***getFrequencyOf(int num)*** – returns a count the number of times the number ***num*** exists in the bag
- ***contains(int num)*** – Tests whether the bag contains the number ***num***. Returns true when the num is contained in the bag.
- **toString( )** - returns a String of the contents of the bag

- **`equals(Object o)`** **–** returns a true if the parameter o exactly matches the contents of the bag (i.e. same numbers in the same order)

Design a ***java class*** called ***Scores*** that implements the ***Bag interface*** and provides implementation for all the methods inherited from the ***Bag*** interface. Do the following in the ***Scores*** class:

- Declare an instance variable ***list*** – an array of ***int*** type: This structure will hold the numbers in the bag.
- Declare another instance variable ***count*** - ***int*** type: This will provide the count of numbers currently stored in the bag. This count will increment when a new number is added to the list and decrement when a number is removed from the list
- Provide a default constructor that will initialize the instance variable list to a new array of length 50.
- Provide an overloaded constructor that will take an ***int*** value as parameter and initialize list to a new array of that length.
- Implement the ***getCurrentSize( ), isEmpty( ) and Clear( )*** methods using the descriptions provided in the ***Bag*** interface ✓ ✓
- Implement the ***add (int num)*** method using the specification provided in the ***Bag*** interface. This method should be able to add a new number to the end of the list ONLY IF the array is not full.
  - If the array list is full (when the count equals the ***length*** of the array) then, create a new bigger array - ***temp*** with double the length of list array.
  - Copy the contents from list to ***temp*** array in the same order.
  - Assign the reference of ***temp*** to list and set temp to ***null***.
  - Add the new number to the end of the list. ✓ ✓

- Implement ***getFrequencyOf(int num)*** and ***contains(int num)*** methods using the descriptions from the ***Bag*** interface
- Implement ***remove(int num)*** method that removes the first occurrence of the number ***num*** in the list array. If the number num does not exist then the method does not do anything.
  - If removal is successful and number removed is not the last number in the list, then shift the elements by one place to the left in the list (i.e. fill in the hole).
- Implement ***remove ( )*** method that removes a random number from the list array,
  - Use the ***Random*** class from ***java.util*** package to generate pseudorandom index.
  - After the number is removed shift the elements by one place to the left in the list (i.e. fill in the hole)
- Implement a new method (a method not in the interface.) called, ***get(int i)*** that returns the number (the value.) at the ***i***th index position in the list. This method does not remove the number from the list, it just

returns the value at the ith position. If the index is outside the bounds of the array, it generates an ***ArrayIndexOutOfBoundsException.***

if index ≥ count -1
- Throw an exception.

Finally design a ***java class Client*** with the ***main( )*** method that does the following:

- Create an Object of Type ***Scores*** using the overloaded constructor and pass the value 100.
- Use a ***for*** loop to populate the list in Scores object with 100 random numbers between -100 and +100 inclusive. (Use the ***Random*** class from ***java.util*** package to generate pseudorandom numbers). ✓
- Call **toString( )** to print the contents of the ***Scores*** object. ✓
- Call the ***add( )*** method to add the number 86 to the Bag ✓
- Print the current size of the list in the Scores object. ✓
- Call the ***remove( )*** method to randomly remove a number from the Bag ✓
- Get the number at the 75th index position ✓
- Print the frequency that the number returned by the previous step occurs in the Bag ✓
- Call the appropriate overloaded ***remove*** method to remove the first occurrence of number at the 75th index position from the Bag ✓ ✓
- Print the frequency that this number now occurs in the Bag
- Print the frequency of the number 86 ✓
- Check whether the list array in ***Scores*** object contains the number 86. ✓

Use ***JavaDoc*** commenting styles in ***Bag*** interface and ***Scores*** class. Make sure to provide a block comment at the top that provides description of each interface or class and a JavaDoc comment for each method.

Use single line commenting style as needed.

Things to turn in:

- Open a Microsoft Word document named
- Copy and Paste the source code of the ***Bag*** Interface (make sure to use *Ctrl + A* to select all the source code of the program and *Ctrl + C* to copy).
- Copy and Paste the source code of the ***Scores*** class.
- Copy and Paste the source code of the client class.
- Copy and paste the output of the client program
- Next, zip the Java Project folder.
- Finally, on blackboard, submit your Word document, project zipped file.