

|  |
| --- |
| **COMP 1230 – Programming 2**  **Assignment #5 Generics** |

## Due Date: Friday July 29, 2022, 11:20 am

**The purpose of this problem is to practice creating and using a generic Bucket class. You must create you own generic class, not use any existing Java generic class.**

**NOTE: Refer to the code for the ArrayStack class from Chapter 12. Use that code as a starting point to create the Bucket class, modifying it to remove the methods in the ArrayStack class (push, pop, etc) then add the methods described below. Also change the variable names to reflect the new class. For example the array name should NOT be stack, instead it should be contents (as in the contents of the Bucket), etc..**

**The contents of the Bucket MUST be stored in a generic array. An ArrayList must not be used.**

**Create a generic class, called Bucket, with a generic type parameter** that simulates drawing an item at random out of a Bucket. For example the Bucket might contain Strings representing names written on a slip of paper, or the Bucket might contain integers representing a random drawing for a lottery. Include the following methods in your generic class, along with any other methods you’d like:

* an add( ) method that allows the user to add **one** object of the specified type. Attempting to add too many items to the Bucket will throw a FullBucketException to the application
* an isEmpty( ) method (returns true if the Bucket is empty, otherwise returns false)
* a drawItem( ) method that randomly (see Note below) selects an object from the Bucket and returns it. **Delete the selected item – do not store a null in this spot. You must fill this gap by shuffling items one index to the left, or by putting the last item in this spot. Whatever strategy you use, the last filled index of the array will be made null.** Attempting to draw from an empty Bucket will throw an EmptyBucketException to the application
* a toString( ) method (returns a String containing the Bucket’s contents). If the Bucket is empty the toString should return “the Bucket is empty”

Your Bucket class with need an array of size 10 to hold the objects in the Bucket, and a count variable to maintain a count of how many objects are actually in the Bucket.

In the driver file that tests your class create 3 Buckets, one to test all the methods in the Bucket class, one with the names of 5 of your friends, the third with numbers between 2 and 4 inclusive representing the number of hours you will spend playing video games with 1 of your friends.

For **the test Bucket**,

* before you put anything in it, call the isEmpty method, the toString method and the drawItem method and show what output each method produces.
* create a for loop to attempt to add 11 integers to the Bucket. Your code should catch the exception and write out an appropriate message
* call the toString to display the contents of the Bucket after you’ve added 10 items
* create a for loop to attempt to draw 11 items out of the Bucket. **After each draw print out the item that is drawn, and call the toString( ) to display the contents of the Bucket after that draw.** Your code should demonstrate:
  + each item drawn is random
  + each item drawn is different
  + attempting to draw the 11th item should throw an exception. Your code should catch the exception and write out an appropriate message
  + call the toString once the loop is done to display the contents of the now empty Bucket

For the remaining 2 Buckets, use the add( ) method to populate the 2 Buckets, and the drawItem( ) method for each Bucket to determine i) which friend you will play video games with and ii) how many hours you and your friend will be playing video games.

**Remember to include lots of descriptive comments in your output.**

Note:

public int nextInt(int bound) in java.util.Random returns a random int value between 0 (included) and the specified value, bound, (excluded).

**Assignment Submission:**

Hand-in a print-out of each class and the output produced.