Project #1 - Building a magic bag

Learning Objectives

- Apply basic object oriented programming concepts in C++
- Construct and use C++ objects making effective use of references and pointers
- Implement an abstract data type conforming to specific design specifications

Overview

Your task for this assignment is to build a “Magic Bag” data type. You will be given a .h file with function prototypes for the magic bag. You will implement the functions according to the following specifications.

The Magic Bag

The magic bag data type will be implemented as follows:

- Objects can be inserted into the magic bag using the **`MagicBag::insert(item)`** member function. The magic bag can hold any number of items, within the bounds of the available RAM. Duplicate items are allowed.
- Objects are removed from the magic bag using **`the MagicBag::draw()`** member function. This function returns a random item and removes it from the bag. The returned item should be randomized. In other words, if there are 5 items in the bag, and only one of them is a 7, then there should be a 1 in 5 chance of drawing a 7 with the draw() function. If the bag is empty, the draw() function should throw an exception.
- You can check if an object is in the bag using **`MagicBag::peek(item)`**, which should return 0 if there is no item in the bag matching the *item* parameter. If there are items matching the *item* parameter in the bag, the number (count) of matching items should be returned.
- You can print a magic bag using the **`MagicBag::print(ostream&)`** member function. This function should print to any ostream you pass to it (cout, cerr, etc.).
- You should be able to assign the contents of a MagicBag using the = operator. This should result in a copy of the magic bag that is not linked to the original bag. In other words, if *a* and *b* are magic bags, the line "a = b;" should result in bags *a* and *b* having the same contents. If I then draw elements from bag *b*, this should not change the contents of bag *a*.
- You should begin by constructing a magic bag that holds integers. For additional credit, you can create a template so that the magic bag can hold any primitive data type (see below for grading details).
- You can implement the magic bag using an array, a linked, list, or any other data structure that you feel is appropriate. You need to design and implement the underlying data structure yourself. You may not, for example, use C++ standard template library vectors for the underlying data structure.
- You will be provided with a main program for testing your magic bag.
- No points will be awarded for submissions that do not compile.

Turn in and Grading

Please build your class in-line in the MagicBag.h file provided. Please turn in ***only your MagicBag.h file*** to the dropbox. Do not zip, archive, or compress your file.

This Project is worth 50 points, distributed as follows:

Task	**Points**
You can create a magic bag of integers and insert integers into the bag.	5
You can create a magic bag of any primitive data type and insert items into the bag.	5
The capacity of the magic bag is limited only by the amount of available RAM.	5
The peek() member function returns a value without deleting any items from the bag.	5
The draw() member function returns and removes an item from the bag.	5
The probability of drawing an item from the bag is equall for all items.	5
You can print Magic Bags to cout using the print(ostream&) member function.	5
Magic bags can be copied and assigned, resulting in a new, independent bag with the same contents as the original bag.	5
You code has no memory leaks.	5
Your Project is well written, well documented, and well structured.	5