Programming Assignment 1 –
Bag-based Dictionary

Implement a dictionary using a Bag—Project 4.7 in the text (modified)

Use the bag ADT provided to create an array-based (do not use the C++ vector library) implementation for bags. Then use your bag to implement the dictionary ADT provided you.

Test your bag and dictionary implementations with the following template combinations:

* <int, string> -- all functions
* <string, int> -- insert and find functions only

You will only get credit for those methods you test and display, so be sure you don’t leave any out.

Put the following files into a zip file named student\_name\_Bag\_Assignment and submit them to Blackboard:

* ABag.h // Your bag-array implementation
* BDictionary.h // Your Dictionary-bag implementation
* dictionaryTest.cpp // Your test program
* screen\_shots.doc/txt // Word document with screen shot(s) showing
 // program in action or a text file showing the test
 // program's output.
* Approach.doc // This is a Word document where you explain how you implemented
 // your solution and how you tested each required function.
* Your\_dictionary.exe //Your executable file.\*\*\*

\*\*\* If you completed your assignment using Visual Studios you must use Visual Studios 2010 or 2013 and I recommend you run your executable on another computer to insure it will run as a stand-alone executable. Depending on how you initialized your project it may not run in the stand-alone mode. If that is the case, you will have to zip up your entire project directory and submit that to me. See me if you have any questions.

Your test program must exercise every function of the dictionary. For any function whose functionality is not obvious you must explain in your Word document how your test output demonstrates that function. See me if you have questions.

Your assignment is due Wednesday of Week 4. **Late assignments will not be accepted.**

Rubrics:

* **Program must run in order to get any points. By “run” I mean that at you must at least get one or more of the bag methods working (and your program must demonstrate that functionality).**

Tips for Success

Start by working on your “Approach” first. Once you are satisfied with your approach, then start building your program incrementally. Start with the bag and increment one feature at a time (you’ll have to stub out the features the ADT requires that you are not ready to implement yet) starting with the constructors and then working your way down the feature list using common sense to figure out which features need to be implemented first. Try your bag out with the various parameter combinations I want you to test with (<int, int>, <string, string>, etc.). When you are satisfied the bag is working then move on to the dictionary, again implementing and testing function by function.

Don’t wait until the last minute. You’ll find that many of your problems you will solve while you are away from your computer and have a chance to think about the error you are seeing. This takes time.

Note: KVpair, which uses the == operator for comparing the key values, will only accept objects that have also implemented the == operator. This class has been tested with the following types:

* + - string
		- int

It specifically does not work with the Int type (at least not in the version of C++ I am working with).