Homework 1: Enhancing the Bag ADT
ECE 2574, Spring 2018
Due before 5:00 pm on February 9

Honor Code: You must work *independently* on this assignment. Please review the statement in the syllabus.

Objectives: The main goals of this assignment are to gain familiarity with Abstract Data Types and to get experience with unit testing of code.

Before you begin: Review the Quiz 1 materials again. If you experienced difficulty with Quiz 1, it is best to seek help before continuing with this assignment.

Download the starter code, provided in file `hw1_starter.zip`. This zip archive will contain several files that are similar to those that you received for Quiz 1. CMake will use the file `CMakeLists.txt` to create a Visual Studio project named `hw1_test`. Associated with this project are `BagInterface.h`, `ArrayBag.h`, `catch.hpp`, and `hw1_test.cpp`. Because of the way that templates are handled, the implementation file `ArrayBag.cpp` must not be associated as a Source File; it will show up automatically under External Dependencies.

You should not modify `catch.hpp`. You should modify all of the other `*.h` and `*.cpp` files as needed to satisfied the project specification.

Project specification:
1) Modify the ArrayBag module to add a new method called `replace`. The prototype and description are as follows:

```
/** Replaces all occurrences of anEntry in this bag with the
    user-provided newEntry. If anEntry is not present, makes
    no changes to the bag and returns false.
 @post  If successful, all occurrences of anEntry will be
    changed to the value newEntry.
 @param anEntry  The entry to be replaced, if already present
    in the bag.
 @param newEntry A new value that will replace all
    occurrences of anEntry in the bag.
 @return  True if at least one occurrence was replaced;
    otherwise return false. */
virtual bool replace(const ItemType& anEntry,
                     const ItemType& newEntry) = 0;
```

As an example, suppose that bag `b` contains the integers {2, 9, 17, 9, 17, 9}. The call

```
b.replace(17, 30);
```

will return `true` and will cause `b` to contain {2, 9, 30, 9, 30, 9}. The call

```
b.replace(5, 30);
```

will return `false` and will make no changes to `b`, because `b` does not contain any occurrences of 5.

2) Modify the ArrayBag module to add a parameterized constructor that creates a bag from a given array of entries. The prototype of this new constructor is as follows:

```
ArrayBag(const ItemType arr[], int size);
```

Additional requirements: At the beginning of your test file `hw1_test.cpp`, place a comment header similar to the following:

```
////////////////////////////////////////////////////
// ECE 2574, Homework 1, Jane Doe ← your name here
//
// File name:   hw1_main.cpp
// Description: Test a modified bag ADT.  (More description here)
// Date:        2/5/2018  ← completion date here
//
```

You should also add appropriate comments to any changes that you make to the other files.

Online testing: We will use the INGInious autograding system to determine part of your grade for this assignment. To use the autograder, place the following C++ source files into a single zip file: `BagInterface.h`, `ArrayBag.h`, `ArrayBag.cpp`, and `hw1_test.cpp`. A suggested file name for the zip archive is `hw1_`*name*`.zip`, where *name* is your family name.

Submit your zip file to the INGInious autograder at https://grader.ece.vt.edu. It will compile and run the tests that you have formulated in `hw1_test.cpp`, and the autograder will also run some "instructor" tests. A grade will be reported to you, proportional to the number of tests that have executed correctly. (One "test" corresponds to a REQUIRE block in the Catch testing framework.)

You can submit to the autograder as many times as you like, but it will limit you to 4 submissions every hour. (This limit is to prevent people from using it as their only compiler.) An advantage of autograding is that you will have peace of mind concerning the major portion of the grade for this assignment.

Submission to Canvas: After you are satisfied with your code based on your autograde results and addressing the requirements, upload the same a zip file (as described above) to Canvas at the HW1 assignment link. You do not need to submit any other files or directories.

Be careful to verify that you have uploaded the correct files to Canvas. After you have uploaded your zip file, it is suggested that you also download it from Canvas and verify that it is correct.

Grading: There are 100 points allocated to this assignment, and most of the grade will be determined by the autograder.

- Correctly submitting the required files to INGInious and to Canvas: 5 points
- Your tests compile: 5 points
- Your tests pass: 20 points (proportional)
- Instructor's tests compile with your code: 5 points
- Instructor's tests pass: 50 points (proportional)
- No memory leaks (checked by autograder using `valgrind`): 5 points
- Good coding style, and your tests are reasonably thorough: 10 points (assessed manually by GTA)

Please note that your most recent submission to INGInious determines the autograded portion of your grade.