Assignment 2 - Due February 28th 11:45 PM

Assignment # 2

Container Classes

Make sure you have read and understood

- ***lesson module week 4***
- ***chapter 3 of our text***
- ***Coding Style Guidelines (module week 1)***

before submitting this assignment. Hand in only one program, please.

Background:

In many applications, the composition of a collection of data items changes over time. Not only are new data items added and existing ones removed, but data items may be duplicated in a collection. The following ADT will support these operations using an array implementation. *Do not use an STL class for this assignment.*

Objective:

Design and implement a container class.

Requirements:

Design a C++ container class to hold a collection of items.

Precondition: The container is initially empty.

Postcondition: The class implementation passes the **Test Run Requirements**.

Understand the Application

An **Bag** object provides a container that supports the following operations on the bag_type data items.

Functional requirements:

Inserts an item in the bag

Determines if an item is in the bag

Determines the number of copies of an item in the bag

Determines the total number of items in the bag

People

Participants

Activities

Assignments
Forums
Quizzes
Resources

Administration

Course administration

Quickmail

Compose New Email
Signatures
View Drafts
View History

Removes an item from the bag

Removes all items from the bag

The Program Spec

Implement a Bag container class that holds a collection of bag_type data items supporting the operations as specified above in the functional requirements.

Test Run Requirements: Only submit one run that shows a sample client that instantiates a Bag object and displays the test cases shown below:

Test run:

Insert a “four” into the bag

Fill the bag to capacity

Show the size of the bag

Display the number of “fours” in the bag

Remove a four from the bag

Show the size of the bag

Display the number of fours in the bag

Display the number of fives in the bag

Remove all fives in the bag

Display the number of fives in the bag

Paste a copy of your test run output display as a multi-line comment (i.e. use delimiters /* */ to encase your run) at the bottom of your test driver file.

Note: For purposes of demonstration set the bag CAPACITY to 20 items.

Grading Criteria:

Bag container class is correctly defined and implemented. Use a typedef to specify the bag_type data items to be collected.

Implementation supports the operations given in the functional requirements.

Mutators filter parameter data.

A constructor is specified to fulfill the precondition design requirement.

A test driver is included to satisfy the postcondition requirement.

Program compiles and runs.

A copy of your test run output display is included as a multi-line comment (i.e. use delimiters /* … */ to encase your run) at the bottom of your test driver file.

Be sure to include 3 separate files:

- bag.h

- bag.cpp
- bagDr.cpp

Submission status

Submission status	No attempt
Grading status	Not graded
Due date	Tuesday, February 28, 2017, 11:45 PM
Time remaining	7 days 10 hours
Last modified	-
Submission comments	Comments (0)

Add submission

Make changes to your submission