Homework 2

Due 30 January 2022, 11:59pm

**Name**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**What to submit**:

* For text response: Use the provided boxes to answer problems that require text response. Save this file as a pdf, and upload this file to Gradescope. If you use extra pages, write “To Be Continued” in the answer box.
* For programming: For each programming question, run `make submission` to generate a zip file and upload it into Canvas.

**Questions**:

1. (text response) A matrix is a rectangular array of numerical values commonly used in signal and image processing algorithms. You may recall that you can add or multiply two matrices to form a third matrix. You can multiply a matrix by a scalar, and you can transpose a matrix. **Design an abstract data type that represents a matrix that has these operations.** Specify each ADT operation by stating its purpose, by describing its parameters, and by writing a pseudocode version of its header. Then write a C++ interface for the methods of the matrix. Include javadoc-style comments in your code (see Carrano p. 759).

Your answer.

1. (text response) When shopping online, you select items and add them to a shopping cart. Duplicate items are permitted, as you can purchase multiples of an item. You can also remove an item from a shopping cart if you change your mind about buying it. The shopping cart can show its current contents with their prices and the total cost of these items. Design the **ADTs** for **ShoppingCart** and **Item**. Then, define classes **Clothing** and **Electronics** as **subclasses** of Item.  
     
   **Hint:** This is an open-ended problem, how complicated the design is up to you. What we are looking for is well-documented class **declarations** with Javadoc style comments, descriptions of all member variable and methods used, stating all the assumptions. You do not need to implement the classes.

Your answer.

1. (programming) Create an interface for the generic Bag ADT and adapt our implementation of Bag to use this interface definition.
   1. Download updated starter code files provided on Canvas (bag\_simple.hpp and CMakeLists.txt) and copy them into your VM repository in the folder hw02/bag\_interface/, replacing the existing files.
   2. In the file abstract\_bag.hpp define a C++ **interface** for our Bag ADT. Have a look at the bag\_simple.hpp to know the functions needed to be defined in the abstract bag class.

More specifically, define an abstract class called ‘AbstractBag’ in abstract\_bag.hpp. This abstract class should have **only pure virtual methods** corresponding to methods of ‘Bag’ class.

* 1. Adapt the Bag implementation using in the files bag\_simple.hpp and bag\_simple.tpp to use this interface in abstract\_bag.hpp. (The bag class should be derived from the abstract bag class).
  2. Build your code locally as you work and make sure that it passes the test cases found in bag\_tests.cpp

Extra Pages

[Write “To Be Continued” in the above answer box if you use this extra space]

Extra Pages

[Write “To Be Continued” in the above answer box if you use this extra space]