CS 300 Program #2

Due: 09/26/2016 11:59PM. *Cut off: 09/28/2016 11:59PM.*

Points possible: 100

General Description

Improve program #1 by:

- Set all `class Part` member variables private.
- Design and implement `class Part` constructors, setters & getters.
- Design `class ArrayList` and use its object to replace `array[100]`.
- In `class ArrayList`, support `delete_part()` to delete a part.
- Separate your program into header files and cpp files.

Data Structure

Define a `class Part` that will hold:

1) `part_number` (integer), from 1 to 100.
2) `description` (string), which allows spaces, e.g. “a useful tool”
3) `quantity` (integer),
4) `price_per_item` (float).
5) Default constructor and custom constructor.
6) Setter and getter functions to set and get variable value, e.g.
    a. `setPartNumber()` and `getPartNumber()`
    b. `setDescription()` and `getDescription()`
    c. setters and getters for `quantity` and `price_per_item`.
7) A `printInformation()` function that prints all member variables.
8) A `subTotalValue()` function that returns `quantity * unitPrice`.

Define a `class ArrayList` that will hold:

1) `int array[CAPACITY]`, where `CAPACITY` is defined as `100`
2) `int count;`
3) Default constructor.
4) Setter and getter function for element in array specified by `index`.
5) Getter for `count`. Note there is no setter for `count`.
6) `add()` function, which adds element after all elements already in the array, if possible (not exceeding `CAPACITY`)
7) `delete(index)` function, which delete the element at `index`, if possible. Note that we cannot have any holes in the array, so move following elements to fill the hole caused by deletion.

Program Function

Use “inventory.txt” to keep inventory record on disk. Initially there is no such file but it should exist after first run, provided some parts have been added to inventory.

The program should have a menu that:

1) `print parts`, which prints all parts in inventory.
2) `enter a new part`, which allows use to enter a new part info.
3) `modify a part`, which searches and modify part infoby part number.
4) `delete a part`, which deletes the part by part number.
5) `print total`, which print the total price of whole inventory.
6) `exit the program`, which writes the inventory to disk and exit program.

Requirements

Read program requirement document carefully.
Build your program INCREMENTALLY.
Use a `class ArrayList` object in `main()` to hold all objects of `part`.
Read `inventory.txt` file only once for each run of program.
Write `inventory.txt` file only once for each exit of program.
Design `inventory.txt` by your self for your program.

Sample Output

Sample output if 1 selected

```
     1    hammer    2    $  10.00
     2    drill press    1    $100.00
```

Sample input if 2 selected

```
     Enter item:  block sander
     Enter quantity:  3
     Enter cost: $50.00
```

Sample input if 3 selected

```
     Enter part to modify: 1
     1    hammer    2    $  10.00

     Enter item:  hammer
     Enter quantity:  3
     Enter cost: $10.00
```

Coding Style Requirement

1) Put your name and myWSU in each source file (as comment).
2) Put comment before each function, explaining purpose of function and preconditions (if any).
3) Put comment in an elegant and readable way.
4) Use indentation and blank lines to format your code.
5) Name each variable, function and file properly.

How to Submit

Use command `~cs300n/bin/handin 2 <file_1.h file_2.cpp … >`

`~cs300n/bin/handin`	: this is the handin program
`2`	: this is the assignment number
`<file_1.h file_2.cpp … >`	`: list all needed files`

Before due time, you can submit as many times as necessary. New submission will overwrite previous ones.