

## Programming Assignment 6: Inventory Class Objects List

CGS 3406 - Spring Term 2010

Point Value: 100 points

Project Due : Wednesday April 21, 2010 at 11:59 PM

**Reminder: this assignment is due at this specified time deadline and NO late work will be accepted!!!**

### Learning Objectives

- To write a program which requires the creation of a class data structure
- To write constructors and other member functions for the class, including accessor and mutator (get and set) functions
- To write an application program for the class to demonstrate its use with an array of class objects
- To use the typedef construct to set up a data type for an array of class objects

### Problem Statement

Create an inventory class that can hold information and calculate data for items in a retail store's inventory.



The *Inventory* class must have the following private data members:

Member	Description
itemName	a C++ string class object
quantity	an int, quantity of items on hand
	a double; wholesale per-unit cost of

cost	the item
totalCost	a double; the total inventory cost of the item (calculated as quantity <i>times</i> cost)

The class must have the following public member functions:

Member Function	Description
Default constructor	sets all the member variables to zero (to "unavailable" for the string)
Alternate constructor	accepts an item's name, quantity and cost as arguments; copies these values to the appropriate member variables and then calls the setTotalCost function; if any of the provided numeric data is negative, the corresponding member data item is set to zero instead
void setItemName	accepts a C++ string argument that is copied to the itemName member variable; no error checking is done
void setQuantity	accepts an integer argument that is copied to the quantity member variable; if the numeric value passed in was negative, quantity is set to zero instead
void setCost	accepts a double argument that is copied to the cost member variable; if the numeric value passed in was negative, the cost is set to zero instead
void setTotalCost	calculates the total inventory cost for the item (quantity times cost) and stores the result in totalCost
string getItemName	returns the value in itemName
int getQuantity	returns the value in quantity
double getCost	returns the value in cost
double getTotalCost	returns the value in totalCost
void displayItem	prints out the information for one inventory item

Write this program in a single C++ program file.

Demonstrate the class by writing an application for it in main and other non-member functions. Note that your application should include functions other than main, which are not part of the class, but are needed to modularize your application.

Your application program must do the following:

1. First, declare two individual class objects. Use the alternate constructor when you declare them, using this data for the first object: "HDTV", 3, 887.88, and this data for the second object: "Rotary Phone", -14, -1.25.
2. Print out the data members of these two class objects.
3. Declare an array of class objects which can store up to a maximum of 7 objects. You must provide for sub-array processing on this array. The default constructor will be called for every

object in this array when it is declared.

4. Print out values for all the data members of all 7 objects in the array, at this initial point, which is right after the array declaration.
5. Open the data file provided to you on the class web site, named *inventory.txt*. Then read in the first line containing the number of objects for which data is provided, and next the actual inventory data. Each item of inventory data is provided using two lines in the file. If the file does not open when your program tries to open it, abort the run as shown in lectures, along with printing an error message for your user.
6. After you have read in the provided file data, print out the number of actual objects provided in the file and the data for each object. At this time you will only print object data for the part of the array that has meaningful inventory information (which may be fewer than 7).

## Input and Other Requirements

The only input is from the data file. You may assume the file data consists of valid data types. You may assume that the number of objects specified in the file is an integer between 1 and 7. You do not need to do any error checking on the item name strings. Error checking for numeric data is as specified above in the member function descriptions.

## Output

- As usual: introduction, echoprinted input, closing termination message, error messages as needed, and any informative messages the user may need or want to see
- Elements described in this write-up
- Follow the course style guidelines

Be creative!

## Miscellaneous

You must use the C++ string class for all string data in this program.

You must set up the class exactly as specified in this write-up, with class members exactly as described. You may not therefore add or delete data members or function members from the class. Members must behave exactly as described in this write-up.

You should have non-member functions other than main in your program, as your program needs to be modular and use functions and parameter passing appropriately. For example, a function in your main program called "PrintHeading" which printed out introductory program output, would be a non-member function.

## Regarding FSU Computer Competency Requirement

This assignment is assessed according to the FSU Computer Competency Requirement. As such students must demonstrate (1) competent use of a discipline-specific software package: Microsoft Visual Studio C++ and (2) the ability to perform simple transactions using the web/Internet: obtaining all of the assignment materials from the class web site and downloading the required file(s), and turning in the completed work correctly to the class web site.

## What File To Turn In and How to Turn In Your Work using Blackboard

Turn in your C++ program source file, which must be named *invClass.cpp*

Be 100% certain you have *thoroughly* read the handout entitled "*Submitting Your Program Assignments Electronically Using Blackboard*" for complete instructions on how to submit and how to **verify your submission** after you submit it. This will help you to avoid losing large numbers of project points due to improper submission issues. This handout is available on the course web site under "Handouts."

---

Last Update: March 15, 2010 5:10 PM, A. Ford Tyson

---