**Name:**

**Advanced Programming in C++**

**Lab Exercise 3/2/2020 Stardate: 73166.67**

In this exercise, you will design an Inventory class. The class that you design should have a test program to thoroughly test the class. Create separate class definition and implementation files. Run your test program to make sure your class meets all specifications. Print out a sample of your output.

Design an Inventory class that can hold information for an item in a retail store’s inventory. The class should have the following private member variables:

*itemNumber* – an int that holds the item’s number

*itemName* – a string that holds the descriptive name of the item

*quantity* – an int that holds the quantity of the item onhand

*cost* – a double that holds the wholesale per-unit cost of the item

*totalCost* – a double that holds the total inventory cost of the item (calculated as quantity times cost)

The class should have the following public member functions:

*default constructor* – sets all member variables to 0 and the null string

*constructor #2* – accepts the items number, name, quantity, and cost as arguments. The function should call other class functions to copy these values to the appropriate member variables and then call setTotalCost function.

*getItemNumber* – returns the value in itemNumber

*getItemName* – returns the value in itemName

*getQuantity* – returns the value in quantity

*getCost* – returns the value in cost

*getTotalCost* – returns the value in totalCost

*printInventory* – prints the contents of the object

The following member functions should be private

*setItemNumber* – accepts int argument that is copied to the itemNumber member variable

*setName* – accepts a string argument that is copied to the itemName member variable

*setQuantity* – accepts an int argument that is copied to the quantity member variable

*setCost* – accepts a double argument that is copied to the cost member variable

*setTotalCost* – calculates the total inventory cost for the item (quantity times cost) and stores the result in the totalCost member variable