**Take Home Program 5 - Due on or before Sunday 10/21**

**Objective:** Objects and classes – Multiple file Implementation

Submit all three files; start the new page for each file

|  |
| --- |
| **Important instructions:**   * *All programs must include comments at the top of your program: your name, course name-section number (e.g. CSIT 839 -26953), program name and the program description in brief.* * *Copy and paste your program code and outputs in Part B of each program.* * *Once it is done, save and submit this word file via Canvas.* |

**1. Inventory.cpp, InventoryMain.cpp** and **Inventory.h (multiple files)**

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 integer that holds the item’s number.
* Quantity – An integer that holds the quantity of the item on hand.
* Cost – A double that holds the wholesale per unit-cost of the item

**Member functions**

* Default constructor – set all member variables to zero
* Constructor #2 (overloaded constructor) – Accept item number, quantity, and cost as parameters. Call setter functions to copy these values into the appropriate member variables.

**Setter functions:**

* setItemNumber – Accepts an integer argument and validate the input. The input must be 0 or greater.
* setQuantity – Accepts an integer argument and validate the input. The input must be 0 or greater.
* setCost - Accepts a double argument and validate the input. The input must be 0 or greater.

**Getter functions:**

* getItemNumber – Return the value in item number
* getQuantity – Return the value in quantity
* getCost – Return the value in cost

**Other function**

* CalTotalCost – Calculate and returns the total cost.
* validateInt(int);
* validateFloat(double);

**Sample Output:**

Demonstrating the default constructor...

Item number: 0

Quantity : 0

Cost : 0.00

Total Cost : 0.00

Demonstrating the overloaded constructor...

Item number: 124

Quantity : 12

Cost : 84.95

Total Cost : 1019.40

Demonstrating the "set" functions...

Item number: 243

Quantity : 50

Cost : 9.50

Total Cost : 475.00

Demonstrating the input validation functions...

Item Number must be 0 or greater. Please re-enter: 100

Quantity must be 0 or greater. Please re-enter: 10

Cost must be 0 or greater. Please re-enter: 9.50

Item number: 100

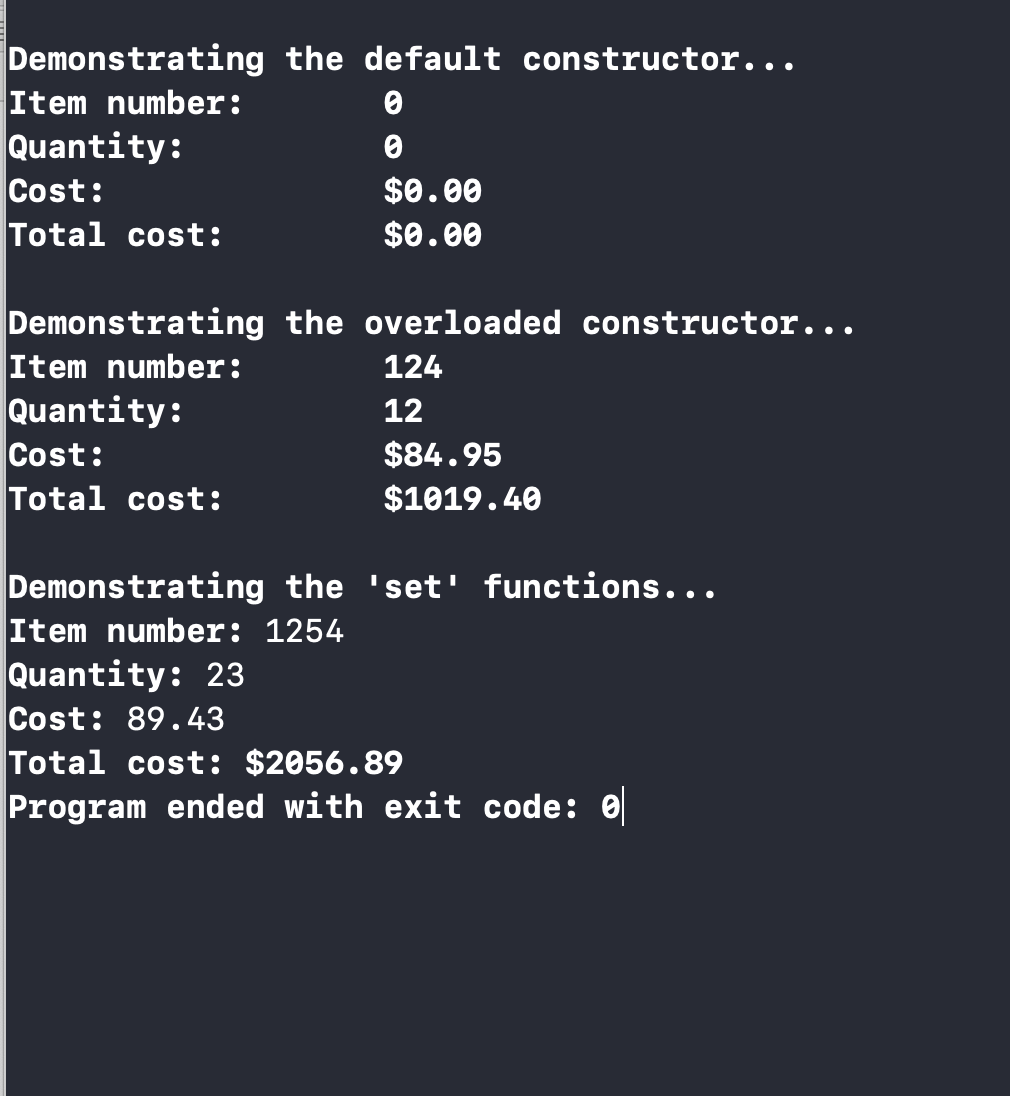
Quantity : 10

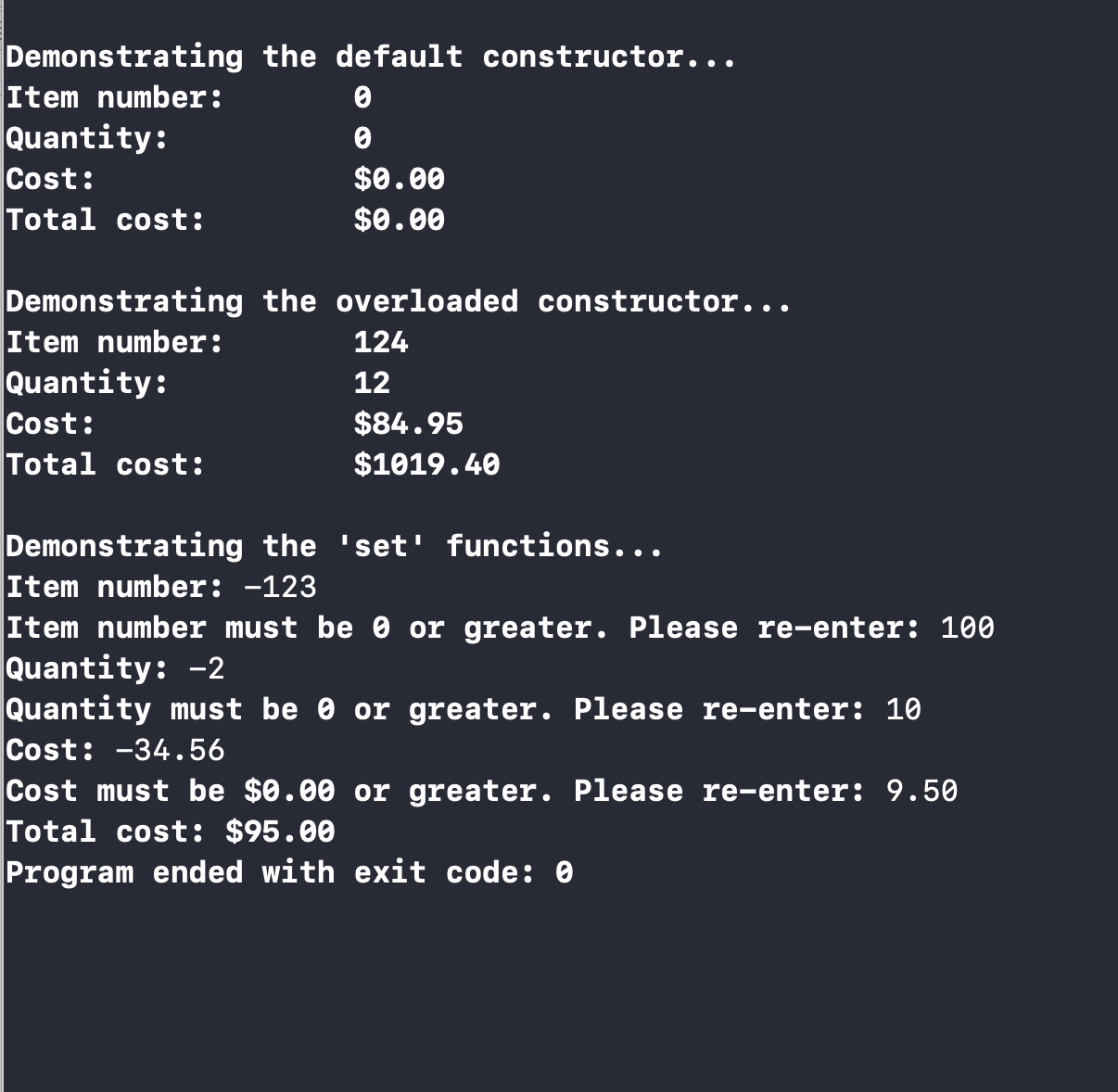
Cost : 9.50

Total Cost : 95.00

**Copy and paste your program (source) code and the outputs after this line**

**Submit all three files; start the new page for each file.**

**+++++++++++++++++++++++++++++++++++++++++++++++++**

****

// Inventory.h - Inventory class specificiation file

#ifndef Inventory\_h

#define Inventory\_h

//Inventory class declaration

class Inventory

{

private:

int itemNumber; //to hold the item's number

int quantity; //to hold the quantity of the item on hand

double cost; //to hold the wholesale per unit-cost of the item

public:

Inventory() //default parameter values

{

itemNumber = 0;

quantity = 0;

cost = 0.00;

}

Inventory(int n, int q, double c) //constructor #2: overloaded constructor

{

itemNumber = n;

quantity = q;

cost = c;

}

//set values of private member variables

void setItemNumber(int);

void setQuantity(int);

void setCost(double);

//get values of private member variables

int getItemNumber();

int getQuantity();

double getCost();

//member functions

double calTotalCost(int, double);

bool validateInt(int);

bool validateFloat(double);

};

#endif /\* Inventory\_h \*/

// Inventory.cpp

//

// Created by Inola Cohen on 10/20/18.

// Copyright ¬© 2018 InolaCohen. All rights reserved.

//

//#include "stdafx.h"

#include <iostream>

#include "Inventory.h"

void Inventory::setItemNumber(int num) {

itemNumber = num;

}

void Inventory::setQuantity(int quant) {

quantity = quant;

}

void Inventory::setCost(double price) {

cost = price;

}

int Inventory::getItemNumber() {

return itemNumber;

}

int Inventory::getQuantity() {

return quantity;

}

double Inventory::getCost() {

return cost;

}

bool Inventory::validateInt(int value)

{

if (value < 0)

{

return false;

}

else

return true;

}

bool Inventory::validateFloat(double cost)

{

if (cost < 0.00)

{

return false;

}

else

return true;

}

double Inventory::calTotalCost(int quantity, double cost)

{

double totalCost = quantity \* cost;

return totalCost;

}

/\*

Inola Cohen

InventoryMain.cpp

CSIT 839 - 26953

Purpose: to design an inventory class

that can hold information for an item

in a retail store's inventory.

\*/

//#include "stdafx.h"

#include <iostream>

#include <iomanip>

#include "Inventory.h" //contains inventory class declaration

using namespace std;

int main()

{

Inventory Inventory1; //declare an inventory object (default -> constructor)

cout << "\nDemonstrating the default constructor..." << endl;

cout << "Item number: " << setw(7)

<< Inventory1.getItemNumber() << endl;

cout << "Quantity: " << setw(10)

<< Inventory1.getQuantity() << endl;

cout << "Cost: " << setw(14) << fixed << "$" << setprecision(2)

<< Inventory1.getCost() << endl;

cout << "Total cost:" << setw(9) << "$" << setprecision(2) << fixed

<< Inventory1.calTotalCost(Inventory1.getQuantity(), Inventory1.getCost()) << endl;

Inventory Inventory2(124, 12, 84.95); //inventory object for overloaded constructor

cout << "\nDemonstrating the overloaded constructor..." << endl;

cout << "Item number: " << setw(9)

<< Inventory2.getItemNumber() << endl;

cout << "Quantity: " << setw(11)

<< Inventory2.getQuantity() << endl;

cout << "Cost: " << setw(14) << fixed << "$" << setprecision(2)

<< Inventory2.getCost() << endl;

cout << "Total cost: " << setw(8) << "$" << setprecision(2) << fixed

<< Inventory2.calTotalCost(Inventory2.getQuantity(), Inventory2.getCost()) << endl;

Inventory Inventory3;

int number, quantity;

double cost;

cout << "\nDemonstrating the 'set' functions... " << endl;

cout << "Item number: ";

cin >> number;

Inventory3.setItemNumber(number);

while (Inventory3.validateInt(number) == false)

{

cout << "Item number must be 0 or greater. Please re-enter: ";

cin >> number;

}

cout << "Quantity: ";

cin >> quantity;

while (Inventory3.validateInt(quantity) == false)

{

cout << "Quantity must be 0 or greater. Please re-enter: ";

cin >> quantity;

}

Inventory3.setQuantity(quantity);

cout << "Cost: ";

cin >> cost;

while (Inventory3.validateFloat(cost) == false)

{

cout << "Cost must be $0.00 or greater. Please re-enter: ";

cin >> cost;

}

Inventory3.setCost(cost);

cout << "Total cost: " << "$" << setprecision(2) << fixed

<< Inventory3.calTotalCost(Inventory3.getQuantity(), Inventory3.getCost()) << endl;

return 0;

}