**WEEK 12 – Review Questions and**

**Programming Challenges Handout**

**Chapter: 6**

**Review Questions:**

**Find the Errors**

1. **Each of the following functions has errors. Locate as many errors as you can.**

**A)** int total(int value1, int value2, int value3)

{

return value1 + value2 + value3;

}

**B)** double average(int value1, int value2, int value3)

{

double average;

average = (value1 + value2 + value3) / 3;

}

**C)** int area(int length = 30, int width)

{

return length \* width;

}

**D)** void getValue(int& value)

{

cout << "Enter a value: ";

cin >> value;

}

**E)** **// Overloaded functions**

int getValue(int& inputValue)

{

int inputValue;

cout << "Enter an integer: ";

cin >> inputValue;

return inputValue;

}

double getValue(double& inputValue)

{

double inputValue;

cout << "Enter a floating-point number: ";

cin >> inputValue;

return inputValue;

}

**Chapter: 7**

**Review Questions:**

2**. Once a class is declared, how many objects can be created from it?**

A) 1

B) 2

C) Many

1. The procedures, or functions, an object performs are called its methods.
2. Bundling together an object’s data and procedures is called encapsulation.

**Programming Challenges:**

**Chapter: 7**

1. **Inventory Class**

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.

The class should have the following public member functions.

**Variable Name Description**

**itemNumber** An int that holds the item’s number.

**quantity** An int that holds the quantity of the item on hand.

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

**Member Function Description**

**default constructor --** Sets all the member variables to 0.

**constructor #2 --** Accepts an item’s number, quantity, and cost as arguments. Calls other class functions to copy these values into the appropriate member variables. Then calls the setTotalCost function.

**Member Functions Description**

setItemNumber Accepts an int argument and copies it into the itemNumber member variable.

setQuantity Accepts an int argument and copies it into the quantity member variable.

setCost Accepts a double argument and copies it into the cost member variable.

getItemNumber Returns the value in itemNumber.

getQuantity Returns the value in quantity.

getCost Returns the value in cost.

getTotalCost Computes and returns the totalCost.

Demonstrate the class by writing a simple program that uses it. This program should validate the user inputs to ensure that negative values are not accepted for item number, quantity, or cost.

**Add totalCost var & setTotalCost function to the Inventory Class**

**In-Class Lab: Calculator Program**

You will be working for this week – it has its own link.