**CSC 3020 – Java Programming**

**Homework 5 – Darryl Green**

**25 points – Due July 12, 10am**

**Late deadline is July 14, 11:59pm, but 20% off**

**a)** Save this document with your name and the homework number somewhere in the file name.

**b)** Type/paste your answers into the document.

**c)** Submit this document and your .java file(s) to the Blackboard item where you downloaded this document. Do not submit a zip file but individually attach your files.

You’ve been hired by *Half Foods* to write a Java console application that manages product inventory. The application has the following two classes:

**Product.java**

This class represents one product in their inventory and includes:

● Fields

✓ (static) productCount – count of all distinct products; initialize to 0 in declaration.

✓ (static) inventoryValue – total inventory value of all products; initialize to 0 in declaration.

✓ (static) inventoryCount – total count of all products in inventory; initialize to 0 in declaration.

✓ code

✓ name

✓ cost

✓ count

● A constructor with no parameters that sets the fields, respectively, to these values:

productCount = productCount + 1

code = -1

name = "(not set)"

cost = -1

count = -1

● A constructor with four parameters that sets the fields, respectively, to these values:

productCount = productCount + 1

inventoryValue = inventoryValue + (cost \* count)

inventoryCount = inventoryCount + count

code – set from parameter

name – set from parameter

cost – set from parameter

count – set from parameter

● Getter methods for each field (declare the getters for productCount, inventoryValue, and inventoryCount static).

● Setter methods for each field (declare the setters for productCount, inventoryValue, and inventoryCount static).

● *equals* method that compares codes from two objects for equality.

● *toString* method for returning instance variable values only.

**HW5.java**

This class contains the main method and uses the other class to manage Product data. Create text file *ProductInventoryIn.txt*, paste the following data into it, and place the file in your project folder. It has the following columns:

1) Product code

2) Product name

3) Product cost

4) Product quantity

**ProductInventoryIn.txt**

**70 Gourmet Popcorn 2.80 50**

**71 Veggie Burgers 5.00 60**

**72 Italian Bread 4.00 70**

**73 Electrolyte Water 2.20 80**

**74 Shaved Parmesan 8.00 90**

**75 Lemonade 1.50 40**

**76 High Protein Bars 1.30 30**

**77 Ginger Shampoo 7.50 20**

**78 Blueberry Toothpaste 3.00 10**

**79 Goodnight Candles 2.50 100**

Read the data from file *ProductInventoryIn.txt* into an array of Product objects called **products**. Present the following menu to the user:

Half Foods Menu

1 – Sell product

2 – Order product

3 – List product inventory

4 – Exit

Enter an option:

Here are what the options do:

● **Sell product –** use a validation loop to prompt for and get from the user the code of the product to be sold (it has to be a valid code). Then use a validation loop to prompt for and get from the user the quantity of the product to be sold. Insure that the quantity is not greater than the current inventory for that product. Update the following fields:

✓ (static) inventoryValue

✓ (static) inventoryCount

✓ count for the product

Print a message in formatted columns including:

✓ Code

✓ Quantity

✓ Revenue from sale

● **Order Product –** use a validation loop to prompt for and get from the user the code of the product to be ordered (it has to be a valid code). Then use a validation loop to prompt for and get from the user the quantity of the product to be ordered. Insure that the quantity is greater than zero. Update the following fields:

✓ (static) inventoryValue

✓ (static) inventoryCount

✓ count for the product

Print a message in formatted columns including:

✓ Code

✓ Quantity ordered

✓ Cost of order

● **List product inventory** shows all product data in formatted columns. It then lists the product count, inventory value, and inventory count.

● **Exit** closes the menu.

Continue to process menu options until the user enter 4. Then write the data to file *ProductInventoryOut.txt\*\*\** in the same layout as the input file. Use these menu options and inputs for your last run:

**Option Code Quantity**

**3**

**1 70 10**

**3**

**2 71 40**

**3**

**1 72 20**

**3**

**2 73 25**

**3**

**1 10,74 100,10**

**3**

**4**

**Hint:** declare a keyboard object as a field (global) and close it at the end of method *main*.

**Product.java**

*[your HW5.java code here]\**

**//==============================================================**

**//Darryl Green**

**//07/12/2017**

**//CSC 3020**

**//Oulette**

**// Title: Inventory Manager**

**// Description:**

**// This application manages product inventory**

**// using two application classes**

**//**

**//==============================================================**

**// Define a class**

**public class Product**

**{**

**// define a required variables**

**static int ProductCount=0;**

**static double InventoryValue=0;**

**static int InventoryCount=0;**

**private int Code;**

**private String Name;**

**private double Cost;**

**private int Count;**

**// Define a Method**

**public Product()**

**{**

**ProductCount = ProductCount + 1;**

**Code = -1;**

**Name = "(not set)";**

**Cost = -1;**

**Count = -1;**

**}**

**// Define a Method**

**public Product(int C,String N, double cost,int Ctr)**

**{**

**ProductCount = ProductCount + 1;**

**InventoryValue = InventoryValue + (Cost \* Ctr);**

**InventoryCount = InventoryCount + Ctr;**

**Code=C;**

**Name=N;**

**Cost=cost;**

**Count =Ctr;**

**}**

**public static int GetProductCount()**

**{**

**return ProductCount;**

**}**

**public static double GetInventoryValue()**

**{**

**return InventoryValue;**

**}**

**public static int GetInventoryCount()**

**{**

**return InventoryCount;**

**}**

**public int GetCode()**

**{**

**return Code;**

**}**

**public String GetName()**

**{**

**return Name;**

**}**

**public double GetCost()**

**{**

**return Cost;**

**}**

**public int GetCount()**

**{**

**return Count;**

**}**

**public static void SetProductCount(int prd )**

**{**

**InventoryCount = prd;**

**}**

**public static void SetInventoryValue(double inv )**

**{**

**InventoryValue=inv;**

**}**

**public void SetInventoryCount(int inc )**

**{**

**InventoryCount = inc;**

**}**

**public void SetCode(int cd )**

**{**

**Code = cd;**

**}**

**public void SetName(String N )**

**{**

**Name= N;**

**}**

**public void SetCost(double c )**

**{**

**Cost = c;**

**}**

**public void SetCount(int c )**

**{**

**Count = c;**

**}**

**public boolean Equals(Product prd)**

**{**

**if(Code==prd.Code)**

**return true;**

**return false;**

**}**

**public String ToString()**

**{**

**return +Code + "\t"**

**+Name + "\t"**

**+Cost + "\t"**

**+Count;**

**}**

**}**

**HW5.java**

*[your HW5.java code here]\**

**//==============================================================**

**//Darryl Green**

**//07/12/2017**

**//CSC 3020**

**//Oulette**

**// Title: Inventory Manager**

**// Description:**

**// This application manages product inventory**

**// using two application classes**

**//**

**//==============================================================**

**// Including the header files**

**import java.io.File;**

**import java.io.FileNotFoundException;**

**import java.util.Scanner;**

**public class HW5**

**{**

**// Declare a array variable**

**static Product prd[] = new Product[20];**

**public static void PerformOperation(int Op,int num )**

**{**

**Scanner scan = new Scanner(System.in);**

**}**

**// Implement a method**

**public static void DisplayMenu()**

**{**

**System.out.println("Half Foods Menu ");**

**System.out.println("1 - Sell Product ");**

**System.out.println("2 - Order Product ");**

**System.out.println("3 – List Product I");**

**System.out.println("4 – Exit");**

**System.out.println("Enter Your Choice:");**

**}**

**// Define a main program**

**public static void main(String args[])**

**{**

**int i=0, Code,numb,Options;**

**double Cost;**

**String Name;**

**// Read a text file**

**File fileLoc = new File("H:/Documents/Wayne State University/Summer 2017/CSC 3020/HOMEwork05/ProductInventoryIn.txt");**

**try**

**{**

**Scanner scan = new Scanner(fileLoc);**

**Scanner scan1= new Scanner(System.in);**

**// checks loop condition**

**while(scan.hasNextLine())**

**{**

**Code = scan.nextInt();**

**Name = scan.next();**

**while(!scan.hasNextDouble())**

**Name= Name +" "+scan.next();**

**Cost = scan.nextDouble();**

**numb = scan.nextInt();**

**prd[i] = new Product(Code,Name,Cost,numb);**

**i++;**

**}**

**DisplayMenu();**

**Options = scan1.nextInt();**

**int n=10;**

**while(Options!=4)**

**{**

**int Qty;**

**if(Options==1)**

**{**

**System.out.println("Enter The Code");**

**Code = scan.nextInt();**

**while(true)**

**{**

**for(i=0;i<prd.length;i++) {**

**if(prd[i].GetCode()==Code)**

**break;**

**}**

**if(i>=n)**

**{**

**System.out.println("Enter Valid Code");**

**Code = scan.nextInt();**

**continue;**

**}**

**break;**

**}**

**System.out.println("Enter The Quantity");**

**Qty = scan.nextInt();**

**while(Qty>prd[i].GetCount())**

**{**

**System.out.println("Enter Valid Quantity");**

**Qty = scan.nextInt();**

**}**

**double dbl;**

**int ic;**

**dbl=prd[i].GetInventoryValue() - (Qty\* prd[i].GetCost());**

**ic=prd[i].GetInventoryCount() - Qty;**

**numb=prd[i].GetCount() - Qty;**

**prd[i].SetInventoryValue(dbl);**

**prd[i].SetInventoryCount(ic);**

**prd[i].SetCount(numb);**

**System.out.println("Code - "+Code);**

**System.out.println("Quantity - "+Qty);**

**System.out.println("Revenue From Sale - $"+Qty\* prd[i].GetCost());**

**}**

**else**

**if(Options==2)**

**{**

**System.out.println("Enter The Code");**

**Code = scan.nextInt();**

**while(true)**

**{**

**for(i=0;i<prd.length;i++) {**

**if(prd[i].GetCode()==Code)**

**break;**

**}**

**if(i>=n)**

**{**

**System.out.println("Enter Valid Code");**

**Code = scan1.nextInt();**

**continue;**

**}**

**break;**

**}**

**System.out.println("Enter The Quantity");**

**Qty = scan.nextInt();**

**while(Qty>prd[i].GetCount())**

**{**

**System.out.println("Enter Valid Quantity");**

**Qty = scan.nextInt();**

**}**

**double dbl1;**

**int ic;**

**dbl1=prd[i].GetInventoryValue() +(Qty\* prd[i].GetCost());**

**ic=prd[i].GetInventoryCount() +Qty;**

**numb=prd[i].GetCount() +Qty;**

**prd[i].SetInventoryValue(dbl1);**

**prd[i].SetInventoryCount(ic);**

**prd[i].SetCount(numb);**

**System.out.println("Code - "+Code);**

**System.out.println("Quantity Ordered - "+Qty);**

**System.out.println("Cost Of Order - $"+Qty\* prd[i].GetCost());**

**}**

**else**

**{**

**for(i=0;i<prd.length;i++) {**

**System.out.println(prd[i].toString());**

**}**

**System.out.println("Inventory Count - "+prd[i].InventoryCount);**

**System.out.println("Inventory Value - "+prd[i].InventoryValue);**

**System.out.println("Product Count - "+prd[i].ProductCount);**

**}**

**DisplayMenu();**

**Options = scan.nextInt();**

**}**

**}**

**catch(Exception e)**

**{**

**System.out.println(e);**

**}**

**}**

**}**

**Program output**

*[your program output here (just the last two screens)]\*\**

**

**ProductInventoryOut.txt**

*[your ProductInventoryOut.txt file here]\*\**

70 Gourmet Popcorn 2.80 50

71 Veggie Burgers 5.00 60

72 Italian Bread 4.00 70

73 Electrolyte Water 2.20 80

74 Shaved Parmesan 8.00 90

75 Lemonade 1.50 40

76 High Protein Bars 1.30 30

77 Ginger Shampoo 7.50 20

78 Blueberry Toothpaste 3.00 10

79 Goodnight Candles 2.50 100

\* **Copying-and-pasting application code to a Word document**

1) From the program editor window, press **CTRL-A** and press **CTRL-C**.

2) From within the Word document, press **CTRL-V**.

\*\* **Copying-and-pasting application output to a Word document**

1) From the Eclipse main screen, maximize the Console window.

2) From the Console window, press **ALT-PrintScreen**.

3) From within the Word document, press **CTRL-V**.

\*\*\* Normally, the input and output file are the same. Using two different files makes it easier to test your application (you’re welcome).