**CSE 1302J - Lab 2**

**Part A: A Shopping Cart using a Class defined by the User**

In this exercise you will complete a class that implements a shopping cart as an array of items. The file *Item.java* contains the definition of a class named *Item* that models an item one would purchase. An item has a name, price, and quantity (the quantity purchased). The file *ShoppingCart.java* implements the shopping cart as an array of Item objects.

1. Complete the *ShoppingCart* class by doing the following:

a. Declare an instance variable *cart* to be an array of Items and instantiate *cart* in the constructor to be an array holding *capacity* Items.

b. Fill in the code for the *increaseSize* method. Your code should be similar to that in CDCollection Class – (handout given out in class) but instead of doubling the size just increase it by 3 elements.

c. Fill in the code for the *addToCart* method. This method should add the item to the cart and update the *totalPrice* instance variable (note this variable takes into account the quantity).

d. Compile your class.

1. Write a program that simulates shopping. The program should have a loop that continues as long as the user wants to shop. Each time through the loop read in the name, price, and quantity of the item the user wants to add to the cart. After adding an item to the cart, the cart contents should be printed. After the loop print a "Please pay ..." message with the total price of the items in the cart. Your main method here will be very similar to the code in Part B.
2. See the CDCollection Handout for help with coding *addToCart (…)* and *increaseSize()* methods. You will also have to add a *getTotalPrice()* method to *ShoppingCart* class in order to print the final total at the end of the grocery list (“Please pay …”).
3. Add 9 items to the cart. There should be a subtotal printed after each new item is added to the cart. Turn in *ShoppingCart* class, *Shop* class, and output. Do not turn in the Item class – no changes are made to that code.

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Item.java

//

// Represents an item in a shopping cart.

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import java.text.NumberFormat;

public class Item

{

private String name;

private double price;

private int quantity;

// -------------------------------------------------------

// Create a new item with the given attributes.

// -------------------------------------------------------

public Item (String itemName, double itemPrice, int numPurchased)

{

name = itemName;

price = itemPrice;

quantity = numPurchased;

}

// -------------------------------------------------------

// Return a string with the information about the item

// -------------------------------------------------------

public String toString ()

{

NumberFormat fmt = NumberFormat.getCurrencyInstance();

return (name + "\t" + fmt.format(price) + "\t" + quantity + "\t"

+ fmt.format(price\*quantity));

}

// -------------------------------------------------

// Returns the unit price of the item

// -------------------------------------------------

public double getPrice()

{

return price;

}

// -------------------------------------------------

// Returns the name of the item

// -------------------------------------------------

public String getName()

{

return name;

}

// -------------------------------------------------

// Returns the quantity of the item

// -------------------------------------------------

public int getQuantity()

{

return quantity;

}

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// ShoppingCart.java

//

// Represents a shopping cart as an array of items

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import java.text.NumberFormat;

public class ShoppingCart

{

private int itemCount; // total number of items in the cart

private double totalPrice; // total price of items in the cart

private int capacity; // current cart capacity

// -----------------------------------------------------------

// Creates an empty shopping cart with a capacity of 5 items.

// -----------------------------------------------------------

public ShoppingCart()

{

capacity = 5;

itemCount = 0;

totalPrice = 0.0;

}

// -------------------------------------------------------

// Adds an item to the shopping cart.

// -------------------------------------------------------

public void addToCart(String itemName, double price, int quantity)

{

}

// -------------------------------------------------------

// Returns the contents of the cart together with

// summary information.

// -------------------------------------------------------

public String toString()

{

NumberFormat fmt = NumberFormat.getCurrencyInstance();

String contents = "\nShopping Cart\n";

contents += "\nItem\t\tUnit Price\tQuantity\tTotal\n";

for (int i = 0; i < itemCount; i++)

contents += cart[i].toString() + "\n";

contents += "\nTotal Price: " + fmt.format(totalPrice);

contents += "\n";

return contents;

}

// ---------------------------------------------------------

// Increases the capacity of the shopping cart by 3

// ---------------------------------------------------------

private void increaseSize()

{

}

}

**Part B: A Shopping Cart Using the ArrayList Class (not optional)**

In this exercise you will implement a shopping cart using the ArrayList class. The file *Item.java* contains the definition of a class named *Item* that models an item one would purchase (this class was used in Part A). An item has a name, price, and quantity (the quantity purchased). The file *Shop.java* is an incomplete program that models shopping.

1. Complete Shop.java as follows:

a. Declare and instantiate a variable *cart* to be an empty ArrayList.

b. Fill in the statements in the loop to add an item to the cart and to print the cart contents (using the default *toString* in the ArrayList class). Comments in the code indicate where these statements go.

c. Compile your program and run it.

2. You should have observed two problems with using the default printing for the cart object: the output doesn't look very good and the total price of the goods in the cart is not computed or printed. Modify the program to correct these problems by replacing the print statement with a loop that does the following:

a. gets each item from the cart and prints the item

b. computes the total price of the items in the cart (you need to use the *getPrice* and *getQuantity* methods of the Item class). The total price should be printed after the loop.

3. Compile and run your program. Turn in source and output – no Item class.

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Shop.java

//

// Uses the Item class to create items and add them to a shopping

// cart stored in an ArrayList.

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import java.util.ArrayList;

import java.util.Scanner;

public class Shop

{

public static void main (String[] args)

{

ArrayList<Item> cart = new ArrayList<Item>();

Item item;

String itemName;

double itemPrice;

int quantity;

Scanner scan = new Scanner(System.in);

String keepShopping = "y";

do

{

System.out.print ("Enter the name of the item: ");

itemName = scan.next();

System.out.print ("Enter the unit price: ");

itemPrice = scan.nextDouble();

System.out.print ("Enter the quantity: ");

quantity = scan.nextInt();

// \*\*\* create a new item and add it to the cart

// \*\*\* print the contents of the cart object using println

System.out.print ("Continue shopping (y/n)? ");

keepShopping = scan.next();

}

while (keepShopping.equals("y"));

}

}

Both outputs should have a running subtotal and a final total repeated at the bottom. For three items, the output would look like this:

Item Unit Price Quantity Total

Eggs 1.50 2 3.00

3.00

Item Unit Price Quantity Total

Eggs 1.50 2 3.00

Bread 1.25 4 5.00

8.00

Item Unit Price Quantity Total

Eggs 1.50 2 3.00

Bread 1.25 4 5.00

Milk 2.50 1 2.50

10.50

Please pay $10.50

**Run with 9 items added to cart**