**5 Stars LunchMart**

*** * * * ***

The 5 Stars LunchMart sells only 5 items. These are:

**1** Chips - $1.00
**2** Big Gulp - $2.20
**3** Curly Fries - $3.25
**4** Ham Sandwich - $4.50
**5** Bacon Cheeseburger - $6.00

(It's **not** a very healthy restaurant! :-)

Write a program that enables a customer to select one or more items from the menu (one at a time) and ultimately receive a request for payment. Your program should

1. display the menu,
2. prompt the user to choose a menu item,
3. keep a running total,
4. ask if they are done after each selection and either prompt for a new selection or move on to calculate a bill
5. calculate the tax (NJ tax rate = 7%) and total price, and lastly
6. request payment from the customer.
    a. You should show the customer the sub-total, tax amount and total due.

You should use a **Boolean flag** to indicate when the user has signaled that they are done with their order so that you can stop prompting them for a selection.

This program will challenge you to combine what you've learned about **branching**, **case statements**, **looping** and **class design**.

**Approach:**

**Part A**

1. Build a **Menu** class that has a **displayMenu** method
2. Build a **MainStore** class that displays a store welcome message, displays the menu from your **Menu** class and then prompts the user for a selection

**Part B**

3. Add a **getItemPrice** method to your **Menu** class (hint: use the case / switch statement)
4. Update your **MainStore** class to use your new **getItemPrice** method to retrieve the price of the item selected by the user
5. Calculate the **Tax** based on New Jersey tax rate and display a **sub-total**, **tax** amount and **grand total** to the user for payment

**Part C**

6. Create a **CashRegister** class that has a **Constructor** method for the class (hint: your Constructor should have 3 *private* variables – **subtotal**, **tax**, and **total** that are all initialized to 0), plus the following additional methods:
    a. **addToSubTotal** – which should take the "price" retrieved in **step 4** above and add it to a running total maintained by your **Cash Register** class.
    b. **getSubTotal** – which should send back the current running subtotal
    c. **getTax** – which should calculate tax on the current running subtotal using NJ taxrate and return the tax amount calculated
    d. **getTotal** – which should call **getTax** and that to the running subtotal to calculate a grand total and return the grand total.

**Note:** copy & paste the code from the **Cash Register class** provided at the PACE-Monmouth website on the **Computer Science** class page

7. Modify the parts of your **MainStore** class from **step 5** above to use your new **CashRegister** class methods for adding to the subtotal and retrieving the subtotal, tax and grand total. Hint: remember that you will have to declare a new instance of your CashRegister class by using:
    **a. CashRegister myRegister = new CashRegister( );**

**Part D**

8. Before calculating **tax** and a **grand total**, ask the user whether or not he or she is done with the order.
    a. Use the **.equals** string function. E.g. answer.equals("y") or answer.equals("yes") where "**answer**" is a variable containing the user's response as retrieved from the keyboard
    b. If not done, ask the user for the **next item** and be sure to keep track of the **running sub-total**
    c. ***Only when done*** should you calculate **tax**, **grand total** and request payment
    d. Use a **While loop** to continue prompting the user and be sure to use a **Boolean variable** to break out of the loop when the user indicates he / she is done

**Part E**

9. Ask the user whether he / she will be paying by: **credit card**, **debit card** or **cash**
    a. **Credit card** – ask the user to enter **credit card number**
    b. **Debit card** – ask the user to enter the **debit card number**, and then ask for the **PIN**. Ask the user if he/she wants any **cash back**
    c. **Cash** – ask the user to enter cash into the cash slot