# C#.NET ASSESSMENT 4

**Task:** Create a shopping cart manager in Java using two classes

**Build Specifications:**

- Clone the classroom repo and create a new Java project within the local cloned repo. (Be sure to find the correct location for your project within your local repo folder.)
- Create a Java class. The first is called **co.grandcircus.GroceryItem** (i.e. package: co.grandcircus, class: GroceryItem) with the following specifications:
    - Include three private member variables:
        i. **name**: a string, the name of the item such as "Apple".
        ii. **price**: a double. (Tip: Although Java has a class for currency that you would use in a professional app, you can just use double in this case to keep it simpler.)
        iii. **count**: an integer that represents how many of this item are in the cart.
    - Create public getters and setters for each. (Note that you won't need to use all three getters and all three setters, but please create all of them anyway in the event you want to add on to this app later.)
    - Create a public constructor that takes all three variables, one for each member variable, and initializes the respective member variables.
    - Create a public method called **printItem** which returns void. This method should print one line of information about the item. (You will want to use the String.format method or string concatenation to format the line.)
- Create a second class. This one is called **co.grandcircus.Cart** and has the following specifications:
    - Include two private member variables:
        i. **name**: a string, the name of the shopper such as Linda.
        ii. **cart**: An ArrayList that contains instances of GroceryItem. You should initialize this with a new, empty ArrayList.
    - You do not need to create getters and setters.
    - Create a constructor that takes as a parameter the shopper's name and fills in the name member accordingly.
    - A method called addItem, which returns void. This takes a GroceryItem instance as a parameter and adds it to the cart.

    - A method called printCart, which returns a void. This method loops through the items in the cart and prints out information about each by calling the item's printItem method. The method also keeps track of how many items are in the cart, and the total price of the cart, which it prints at the end. (Hint: Remember that the GroceryItem class has a count. You'll want to use that count. So if you have two apples, be sure to add two to the total count.)
    - A method called printMostExpensive, which returns void. This method will loop through the items and determine the most expensive single item. (Ignore the count when determining the most expensive.)
- In the App.java's main method,
    - Create a new instance of Cart. Use shopper name "Linda".
    - Add the following items to the cart:
        i. Apple, price 1.00, count 5
        ii. Pepsi, price 2.50, count 2
        iii. Cooler, price 35.00, count 1
        iv. Fritos, price 4.50, count 1
    - Call the printCart method.
    - Call the printMostExpensive method.

**Hints**

- If using String.format, remember %s is for string values, %f is for double values, and %d is for integer values. To format with two digits to the right of the decimal, use the %.2f specifier.
- Don't worry about formatting the doubles to have only two decimals.
- For the printMostExpensive method:
    - Start by creating an empty instance of GroceryItem with its price set to 0. After determining the most expensive item, this method writes out a line that says, "Most expensive item:" Then on the next line it prints out information on the item by calling the item's printItem method.

**Example output**

```
Here is Linda's cart.
Apple: 5 items at $1.00 each. $5.00
Pepsi: 2 items at $2.50 each. $5.00
Cooler: 1 items at $35.00 each. $35.00
Fritos: 2 items at $4.50 each. $9.00
Cart Total: $54.00
Item Count: 10
Most expensive item:
```

Cooler: 1 items at $35.00 each. $35.00.

## GRADING

| REQUIREMENTS | POINTS |
|---|---|
| **1.** GroceryItem has three member variables that use the correct types. | 1 |
| **2.** GroceryItem has six getter/setters methods that function correctly. | 1 |
| **3.** GroceryItem has a constructor that initializes all three member variables. | 1 |
| **4.** GroceryItem has a printItem method that correctly prints out the information. | 1 |
| **5.** Cart has the two member variables and the “cart” member is appropriately initialized. | 1 |
| **6.** Cart has a constructor that initializes the shopper name. | 1 |
| **7.** Cart’s printCart method correctly prints the cart total. | 1 |
| **8.** Cart’s printCart method correctly prints the item count. | 1 |
| **9.** Cart’s printMostExpensive correctly finds the correct item and prints out its information. | 1 |
| **10.** The main method creates a cart and correctly calls both printCart and printMostExpensive. | 1 |
| **TOTAL** | **10** |