

Program #1 (20 points)

Important Dates

1. Due date: Monday, Feb. 8 @ 10pm
2. Grace date: Wednesday, **Feb. 10 @10pm**

Program Description

In this program, you will implement a Bag data structure that can hold up to 8 grocery items. You should be able to add and remove items to/from the Bag, print the list of items in the Bag and check out the grocery items in the Bag. Your program should read and process commands from the standard input, and display the results to the standard output. When the program runs, display "Let's start shopping!", and your program **MUST** be able to process the following commands.

- **A** command—**add** a grocery item to the Bag. For example, `A milk 2.99`, which adds grocery item milk with the price 2.99 to the Bag. If the Bag is full, display "Unable to add, the bag is full!". If the item was successfully added, display "milk added to the bag."
- **R** command—**remove** an item from the bag. For example, `R milk 2.99`, which removes grocery item milk with the price 2.99 from the Bag. If the remove was not successful, display "Unable to remove, this item is not in the bag.", otherwise display "Item removed successfully."
- **P** command—**output** all items in the Bag. If the Bag is empty, display "Unable to print, the shopping bag is empty.", otherwise display "You have the following items in the bag: " followed by the list of items in the Bag.
- **C** command—**check out** the grocery items in the Bag. If the Bag is empty, display "Unable to check out, the shopping bag is empty.", otherwise, display "Checking out the following items: ", then call the **P** command to print out all items, and then display: "Amount paid: 18 dollars." where "18" is the result of calling the **totalAmount()** method of the **ShoppingBag** class, and using **Math.round()** method to round it to a whole number.
- **Q** command—terminate the program and display "Thanks for shopping with us!".

Program Requirement

1. You **MUST** follow the software development ground rules.
2. You must submit the program with 0 differences to pass this course.
3. Program submitted with differences or after **Wednesday, Feb. 10 @10pm** is worth **0 points**!
4. This is an **individual assignment**. Sharing your solution could result in being accused of **plagiarism**. I will ALWAYS run the plagiarism check.
5. You **MUST** use **SE tools** to keep an up-to-date log of the time spent working on this program. You must use the automatic **punch in** and **punch out** when possible. Select the activities and provide specific comments about what you worked on during that period. Up to **-5 points** if this is not done or you do not log all your time or you do not provide specific comments as to what you were working on.
6. Each class must go in a separate file. I will provide **Prog1.java**.
7. If you need help from me, place your project in **K:\Courses\CSSE\changl\cs2430\your_login_id**. So I can access it from my office when you stop by.
8. Your program **MUST** handle bad commands! Use the method **.equalsIgnoreCase()** of the String class to check the commands.
9. Download all Java files in the CSSE course file folder Prog1. You **MUST** finish the following classes, and you **MUST** finish all the methods in the Java files. **-2 points** for each method not finished or not used.

- **GroceryItem class.** This class defines the properties and methods of a grocery item. You CANNOT add other data members to this class, you CANNOT change the signature of the methods, and you CANNOT do I/O in this class. **-2 points** if you did.
- **ShoppingBag class.** This class implement the Bag data structure with the methods required to manipulate the Bag. You CANNOT add other data members to this class, you CANNOT change the signature of the methods, and you CANNOT do I/O in this class, except the **print()** method. **-2 points** if you did.

Additional notes for some of the methods in ShoppingBag class.

- (1) add() – append item to the end of the array, NO NEED to check if it is full here.
 - (2) remove() – remove the first occurrence of the item in the array. Find the index of the item to be removed, set the index to reference the last item (move the last item in the array to fill the slot), and set the reference of the last item to null.
- **Shopping class.** This class is the user interface class that handles **input** commands and **output** messages. Finish the run() method and all private methods. The main method in Prog1.java will call the **run() method** here. You MUST check if the Bag is full or empty in the private methods by calling **.isFull()** or **.isEmpty()** methods. **-2 points** if you didn't use these two methods in this class. In the **checkout()** method, you should **empty the Bag** after displaying the total amount.

Sample Input

```
r meat 2.99
P
C
a meat 5.99
A candy 0.99
A chocolate 2.99
blah blah blah
A shampoo 3.59
r candy 1.99
R candy 0.99
P
A chips 2.59
A DVD 14.59
A onion 1.59
A cheese 2.79
A eggs 3.59
A milk 2.69
P
R DVD 14.00
r DVD 14.59
P
a milk 2.69
P
R onion 1.59
P
R meat 5.99
P
C
P
Q
```

Sample Output

```
Let's start shopping!
Unable to remove, this item is not in the bag.
Unable to print, the shopping bag is empty.
Unable to check out, the shopping bag is empty.
meat added to the bag.
candy added to the bag.
chocolate added to the bag.
Invalid command!
shampoo added to the bag.
Unable to remove, this item is not in the bag.
Item removed successfully.
You have the following items in the bag:
meat::5.99
shampoo::3.59
chocolate::2.99
chips added to the bag.
DVD added to the bag.
onion added to the bag.
cheese added to the bag.
eggs added to the bag.
Unable to add, the bag is full!
You have the following items in the bag:
meat::5.99
shampoo::3.59
chocolate::2.99
chips::2.59
DVD::14.59
onion::1.59
cheese::2.79
eggs::3.59
Unable to remove, this item is not in the bag.
Item removed successfully.
You have the following items in the bag:
meat::5.99
shampoo::3.59
chocolate::2.99
chips::2.59
eggs::3.59
onion::1.59
cheese::2.79
milk added to the bag.
You have the following items in the bag:
meat::5.99
shampoo::3.59
chocolate::2.99
chips::2.59
eggs::3.59
onion::1.59
cheese::2.79
milk::2.69
Item removed successfully.
You have the following items in the bag:
```

meat::5.99
shampoo::3.59
chocolate::2.99
chips::2.59
eggs::3.59
milk::2.69
cheese::2.79
Item removed successfully.
You have the following items in the bag:
cheese::2.79
shampoo::3.59
chocolate::2.99
chips::2.59
eggs::3.59
milk::2.69
Checking out the following items:
cheese::2.79
shampoo::3.59
chocolate::2.99
chips::2.59
eggs::3.59
milk::2.69
Amount paid: 18 dollars.
Unable to print, the shopping bag is empty.
Thanks for shopping with us!