



## COMPUTER SCIENCE 12B (SPRING 2020) PROGRAMMING IN JAVA

### PROGRAMMING ASSIGNMENT 3

#### Program Description:

This assignment consists of two parts.

**Part 1:** will give you extra practice with the concepts we have covered so far. **Modularity in your code is very important, YOU MUST USE STATIC METHODS.**

**Part 2:** will give you practice with defining new types of objects. Turn in two Java classes named `GroceryList` and `GroceryItemOrder`, stored in files named `GroceryList.java` and `GroceryItemOrder.java` respectively. You will also need to implement the client code which interacts with the two classes.

#### Part 1:

Write a program that plays a variation of the game of Mastermind with a user. For example, the program can use pseudorandom numbers to generate a four-digit number. The user should be allowed to make guesses until she gets the number correct. Clues should be given to the user indicating how many digits of the guess are correct and in the correct place and how many digits are correct but in the wrong place.

#### Part 2:

Write a class named `GroceryList` that represents a list of items to buy from the market, and another class named `GroceryItemOrder` that represent a request to purchase a particular item in a given quantity (e.g., four boxes of cookies).

The `GroceryList` class should use an array field to store the grocery items and to keep track of its size (number of items in the list so far). Assume that grocery list will have no more than 10 items. A `GroceryList` object should have the following methods (note you may add more methods to your class)

```
public GroceryList() -- Constructs a new empty grocery list
```

```
public void add (GroceryItemOrder item) -- Adds the given item order to this list if the list has fewer than 10 items.
```

```
public double getTotalCost () -- Returns the total sum cost of all grocery item orders in this list
```

The `GroceryItemOrder` class should store an item quantity and a price per unit. A `GroceryItemOrder` object should have the following methods (note you may add more methods to your class):

```
public GroceryItemOrder(String name, int quantity, double
procePerUnit) -- Constructs an item order to purchase the item with the given name, in the
given quantity, which costs the given price per unit.
```

```
public double getCost() -- Returns the total cost of this item in its given quantity. For
example, four boxes of cookies that cost 2.30 per unit have a total cost of 9.20
```

```
public void setQuantity(int quantity) -- Sets this grocery item's quantity to be
given value.
```

Below is a log of execution that demonstrates how the client code would interact with the objects **(user input underlined)**:

```
My Grocery List (you can only have 10 items in the list)
```

```
Please enter an item to add to the list (item-name quantity price)
Enter "quit" to stop adding items to the list
```

```
banana 3 2.5
```

```
milk 1 3.9
```

```
apple 4 3
```

```
quit
```

```
My list: [3 banana, 1 milk, 4 apple, null, null, null, null, null,
null, null]
```

```
Total grocery amount: $ 23.4
```

```
Total number of items: 8
```

### Guidelines:

For this assignment you should limit yourself to the Java features covered in class so far (lecture 14). Although we will cover other topics while you are working on this assignment, do not use any of those features.

### Grading:

You will be graded on

- **External Correctness:** The output of your program should match exactly what is expected. Programs that do not compile will not receive points for external correctness.
- **Internal Correctness:** Your source code should follow the stylistic guidelines shown in class. Also, remember to include the comment header at the beginning of your program.
- **Comments and Style**

### Submission:

Create a folder containing your Java source code (programs). Compress (zip) the folder and upload it to Latte by the day it is due. For late policy check the syllabus.