L5-L6 Homework: Apothecary

Problem Description

Hello! Please make sure to read all parts of this document carefully.

In this assignment, you'll be using your knowledge of conditionals and Scanners to create a program where potential buyers of potions can query the cost of your potion-making services. To do this, you will create and turn in an Apothecary.java file that runs code in a main method.

Solution Description

You will write a **single file, Apothecary.java**. You **must use at least one if-else statement, one switch statement, and one ternary conditional operator**.

Apothecary.java

The program starts with the message, "Welcome to my apothecary! Please enter your name here: "

- Notice the formatting of the name given in the example outputs.
- The name should have the first letter capitalized and the rest in lowercase regardless of what the user inputs (i.e. niColas --> Nicolas, dorothy --> Dorothy).
    - The input will never have spaces
    - If the user doesn't input any name, you will default to "Apprentice"

The program then welcomes the user and prompts them to indicate which potion they seek and how many of them they need. It should print "Hello [formatted name], which potion do you want me to brew? "

The following are possible options:

- Potion of Clarity
- Elixir of Agility
- Healing Draught
- Elixir of Elemental Power

After the user indicates the potion (which should be matched in a case-insensitive manner), the program should do the following special actions if there is no match:

- If the buyer indicates "Death Poison" (case-insensitive), the program should print "GUARDS!" and finish (there should be no more output or request for input).
- If the buyer inputs anything else that doesn't match, the program should print "I am sorry, I cannot brew that potion." and finish.

If a valid potion is selected, the program will then provide the list of ingredients for a single unit of that potion and ask the buyer how many they want, with the input line: "The [potion name] requires [ingredients]. How many would you like? ". Below is the list of ingredients for each potion:

- Potion of Clarity: 2 Vials of Crystal Dew
- Elixir of Agility: 3 Swift Feathers
- Healing Draught: 1 Phoenix Feather and 2 Vials of Moonlit Dew
- Elixir of Elemental Power: 1 Vial of Moonlit Dew, 3 Lava Stones, and 2 Phoenix Feathers

The user will then input their answer in the same line. If the input, which is always non-empty (but not necessarily a number), is not an integer or is less than 1, you will proceed with brewing one potion, otherwise, you will brew the requested amount.

After that, the program will ask how many quantities of each ingredient the buyer can provide (for the given potion, in the order listed above). For each ingredient in a potion, it will ask "How many [ingredient name in plural] will you provide? ". The user will always input a non-negative integer. If the quantity is lower than the total required amount (which should take into consideration the number of potions that will be brewed), the program will calculate the cost of the missing ingredients (and if it’s equal to or greater than the required amount, it will not add any cost). When the program gives the buyer the final cost of the service (which will be done at the last output line), the cost of missing ingredients will be added. Below is the cost of each ingredient:

- Vials of Crystal Dew: $10 each
- Swift Feathers: $20 each
- Phoenix Feathers: $50 each
- Vials of Moonlit Dew: $15 each
- Lava Stones: $30 each

After that, the program should ask the buyer "Is there anything I should know? ". If the buyer inputs "The order is for the King" (case sensitive), you will apply a 50% discount at the end of the program (and otherwise, there will be no discount).

With the information provided, the program will calculate the cost. The cost is calculated in the following way:

- For regular potions (Potion of Clarity, Elixir of Agility), you will add:
    - A service fee of $10
    - $15 per potion brewed (this cost multiplies if the buyer requested an amount greater than 1)
    - The cost of the missing ingredients
- For advanced potions (Healing Draught, Elixir of Elemental Power), you will add:
    - A service fee of $20
    - $25 per potion brewed
    - The cost of the missing ingredients

After that, apply the 50% discount if applicable for the final cost.

Lastly, the program will print the order. Print "[formatted name], thank you for requesting the [potion name, in plural if amount > 1. to make name plural: use Potions, Elixirs, Draughts]. The cost is [total in U.S. currency format, with $ and 2 decimal digits]."

Your program output should always end in a new line after the last line of output.

Example Input/Output

Example Output 1 – User Input is Bolded. Your program should look exactly like this.

```
Welcome to my apothecary! Please enter your name here: ignacio

Hello Ignacio, which potion do you want me to brew? Potion of clarityY
The Potion of Clarity requires 2 Vials of Crystal Dew. How many would you like? 1

How many Vials of Crystal Dew will you provide? 1

Is there anything I should know? nope

Ignacio, thank you for requesting the Potion of Clarity. The cost is $35.00.
```

Example Output 2 – User Input is Bolded. Your program should look exactly like this.

```
Welcome to my apothecary! Please enter your name here:

Hello Apprentice, which potion do you want me to brew? elixir of elemental power
The Elixir of Elemental Power requires 1 Vial of Moonlit Dew, 3 Lava Stones, and 2 Phoenix Feathers. How many would you like? 3

How many Vials of Moonlit Dew will you provide? 0
How many Lava Stones will you provide? 0
How many Phoenix Feathers will you provide? 1

Is there anything I should know? The order is for the King

Apprentice, thank you for requesting the Elixirs of Elemental Power. The cost is $330.00.
```

Example Output 3 – User Input is Bolded. Your program should look exactly like this.

```
Welcome to my apothecary! Please enter your name here: Siam

Hello Siam, which potion do you want me to brew? death POISON
GUARDS!
```

Rubric

[100] Apothecary.java

- [90] 15 scenarios, each 6pt. Each scenario is scored all-or-nothing, depending on whether the input/output matches the solution
- [10] Other checks
    - [1] Apothecary compiles and has main
    - [3] Apothecary.java has an if-else
    - [3] Apothecary.java has a switch statement
    - [3] Apothecary.java has an use of the ternary conditional operator

We reserve the right to adjust the rubric, but this is typically only done for correcting mistakes.

Allowed Imports

To prevent trivialization of the assignment, you are only allowed to import **java.util.Scanner**, **java.text.NumberFormat**, **java.util.Locale**, and **java.text.DecimalFormat**. You are not allowed to import any other classes or packages.

Feature Restriction

There are a few features and methods in Java that overly simplify the concepts we are trying to teach. For that reason, do not use any of the following in your final submission:

- **var** (the reserved keyword)

Checkstyle

The Checkstyle deduction limit for this assignment is 0 points, not counting Javadoc errors. Refer to the course guide for the use of Checkstyle and review the autograder Checkstyle report on your submissions.

Collaboration

Collaboration Statement

To ensure that you acknowledge a collaboration and give credit where credit is due, we require that you place a collaboration statement as a comment at the top of at least one .java file that you submit. That collaboration statement should say either:

I worked on the homework assignment alone, using only course materials.

or

In order to help learn course concepts, I worked on the homework with [give the names of the people you worked with], discussed homework topics and issues with [provide names of people], and/or consulted related material that can be found at [cite any other materials not provided as course materials for CS 1331 that assisted your learning].

If the autograder encounters issues processing your collaboration statement, begin it with “Collaboration Statement: “.

Allowed Collaboration

When completing homeworks for CS 1331 you may talk with other students about:

- What general strategies or algorithms you used to solve problems in the homeworks
- Parts of the homework you are unsure of and need more explanation
- Online resources that helped you find a solution
- Key course concepts and Java language features used in your solution

You may **not** discuss, show, or share by other means the specifics of your code, including screenshots, file sharing, or showing someone else the code on your computer, or use code shared by others.

Examples of approved/disapproved collaboration:

- approved:
    - "Hey, I'm really confused on how we are supposed to implement this part of the homework.
    - What strategies/resources did you use to solve it?"
- disapproved:
    - "Hey, it's 10:40 on Thursday... Can I see your code? I won't copy it directly I promise"

In addition to the above rules, note that it is not allowed to upload your code to any sort of public repository. This could be considered an Honor Code violation, even if it is after the homework is due.

Turn-In Procedure

Upload the files listed below to the corresponding assignment on Gradecope:

- Apothecary.java

For each submission, you will be able to see the results of a few basic test cases on your code. Each test typically corresponds to a rubric item, and the score returned represents the performance of your code on those rubric items only. If you fail a test, you can look at the output to determine what went wrong and resubmit once you have fixed the issue.

Important Notes

- Non-compiling files will receive a 0 for all associated rubric items
- Do not submit .class files or .jar files
- Test your code in addition to the basic checks on Gradescope
- Submit every file each time you resubmit
- Read the "Allowed Imports" and "Restricted Features" to avoid losing points
- Check on Piazza for a note containing all official clarifications