Homework 03 – 1331 Giveaway

Problem Description

Hello! This HW assignment focuses on **lessons 7 and 8**, but we encourage you to use methods (lesson 9) if you feel comfortable with them. Please make sure to read all parts of this document carefully.

You are holding a giveaway with your items laid out in a grid. Using your Java skills, you have the idea to create a way for people to easily take what they are interested in!

Solution Description

Create a file `Giveaway.java` that allows people to take items. We will provide you with an initial array containing items that are available to for grabs (represented by a 5x5 grid). Create a single Java class called `Giveaway` with a main method header.

- Do **NOT** create any static variables.
- Do **NOT** create a Scanner object outside of the main method.
- Additionally, you must only create one instance of a Scanner object.

2D Array Chart

The following String values must be stored within a 2D array called **inventory.** Create the 2D array with the values from the following chart. Each cell visually represents a space for an item. X represents that no item is currently at the location. For these locations, use "x" (lowercase) to represent empty spaces.

Provided 2D array

x	x	StuffedPython	x	CSalt
hAIrspray	x	x	x	x
x	x	x	JavaBeans	x
x	RustedMetal	SwiftShoes	x	x
FuRniture	x	x	GroovyGear	RadiantRuby

Name your program `Giveaway.java`, and it should work as follows.

Acquiring items:

- Welcome the user and print: "Welcome to the 1331 Giveaway!"
- Prompt and ask the user if they want to take an item: "Would you like to take an item? [Y]es, [N]o, or [E]xit"
    - If the user inputs 'N' (case sensitive), print: "Next person in line!" and prompt the next user the same prompt above. See example.
    - If the user inputs 'E' (case sensitive), print: "Have a good day!" and end the program gracefully (i.e., the program ends normally without any exceptions thrown)

    - If the user inputs 'Y', print the current state of the giveaway 2D array and then proceed to print: "What item are you interested in taking?"
        - At this point, the user should enter coordinates separated by a space where the row should be entered first followed by the column to the item's location.
        - See the next section titled "Item selection" for more details.
    - Otherwise, if the user inputs a character outside of 'Y', 'N', or 'E', loop until the user inputs a proper character by prompting, "Please input 'Y', 'N', or 'E'." followed by the prompt "Would you like to take an item? [Y]es, [N]o, or [E]xit"
- The giveaway program should be in a loop such that different users will be able to take an item until either 'E' is inputted or there are no more items left.
- Once there are no items left, instead of printing the regular prompt for Y/N/E, print: "Sorry, we have no more items!"
    - The program should end gracefully afterwards.
    - Note: in this case, "Have a good day!" should not be printed

Item selection:

- If the user input has an invalid row or invalid column, print: "Location does not exist."
    - Re-prompt the user "What item are you interested in taking?"
- If the location inputted is empty (no item in it), print: "There is no item in this location."
    - Re-prompt the user "What item are you interested in taking?"
- If the location inputted has an item, print: "You successfully took the [Name of the item]!"
- **Please be aware that we are only testing numeric input of the location in the correct format.**
    - 0 3 (correct format)
    - 0 , 3 (incorrect format)
    - 03 (incorrect format)
- After an item has been taken, the grid containing the item should be replaced by "x". Print out the new grid.
    - The program should loop back to the previous prompt, "Would you like to take an item? [Y]es, [N]o, or [E]xit"

Example Output#1 - User input is bolded.

(Your program should look **exactly** like this with the exact spacings and lines! If there is any deviation, you will l**ose points.** For better readability, extra newline characters have been added between prompts. These new lines are optional but encouraged.)

```
Welcome to the 1331 Giveaway!

Would you like to take an item? [Y]es, [N]o, or [E]xit
Y

|x|x|StuffedPython|x|CSalt|
|hAIrspray|x|x|x|x|
```

```
|x|x|x|JavaBeans|x|
|x|RustedMetal|SwiftShoes|x|x|
|FuRniture|x|x|GroovyGear|RadiantRuby|

What item are you interested in taking?
0 5

Location does not exist.

What item are you interested in taking?
0 4

You successfully took the CSalt!

|x|x|StuffedPython|x|x|
|hAIrspray|x|x|x|x|
|x|x|x|JavaBeans|x|
|x|RustedMetal|SwiftShoes|x|x|
|FuRniture|x|x|GroovyGear|RadiantRuby|

Would you like to take an item? [Y]es, [N]o, or [E]xit
Z

Please input 'Y', 'N', or 'E'.

Would you like to take an item? [Y]es, [N]o, or [E]xit
E
Have a good day!
```

Example Output#2 - User input is bolded.

(Your program should look **exactly** like this with the exact spacings and lines! If there is any deviation, you will l**ose points.** For better readability, extra newline characters have been added between prompts. These new lines are optional but encouraged.)

```
Welcome to the 1331 Giveaway!

Would you like to take an item? [Y]es, [N]o, or [E]xit
Y

|x|x|StuffedPython|x|CSalt|
```

```
|hAIrspray|x|x|x|
|x|x|x|JavaBeans|x|
|x|RustedMetal|SwiftShoes|x|x|
|FuRniture|x|x|GroovyGear|RadiantRuby|

What item are you interested in taking?
10 5

Location does not exist.

What item are you interested in taking?
0 0

There is no item in this location.

What item are you interested in taking?
1 0

You successfully took the hAIrspray!

|x|x|StuffedPython|x|CSalt|
|x|x|x|x|x|
|x|x|x|JavaBeans|x|
|x|RustedMetal|SwiftShoes|x|x|
|FuRniture|x|x|GroovyGear|RadiantRuby|

Would you like to take an item? [Y]es, [N]o, or [E]xit
N
Next person in line!

Would you like to take an item? [Y]es, [N]o, or [E]xit
E
Have a good day!
```

Rubric

[100] Giveaway.java

- [20] Correctly updates array
- [10] Starts and exits based on Y/N/E
- [15] Correctly prompts the user
- [25] User correctly takes the item

- [10] Correct display of array
- [20] Correct program behavior and output

Allowed Imports

To prevent trivialization of the assignment, you are only allowed to import

- java.util.Scanner

The Checkstyle cap for this homework assignment is 5. Up to 5 points can be lost from Checkstyle errors.

Feature Restrictions

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

- `var` (the reserved keyword)
- `System.exit`

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].

Allowed Collaboration

When completing homeworks for CS1331 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

Submission

To submit, upload the files listed below to the corresponding assignment on Gradescope:

- Giveaway.java

Make sure you see the message stating "HW03 submitted successfully". From this point, Gradescope will run a basic autograder on your submission as discussed in the next section.

You can submit as many times as you want before the deadline, so feel free to resubmit as you make substantial progress on the homework. We will only grade your last submission: be sure to **submit every file each time you resubmit**.

Gradescope Autograder

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.

The Gradescope tests serve two main purposes:

- Prevent upload mistakes (e.g. non-compiling code)
- Provide basic formatting and usage validation

In other words, the test cases on Gradescope are by no means comprehensive. Be sure to thoroughly test your code by considering edge cases and writing your own test files. You also should avoid using Gradescope to compile, run, or checkstyle your code; you can do that locally on your machine.

Other portions of your assignment can also be graded by a TA once the submission deadline has passed, so the output on Gradescope may not necessarily reflect your grade for the assignment.

Important Notes (Don't Skip)

- Non-compiling files will receive a 0 for all associated rubric items
- Do not submit `.class` 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

Pun of the Week :)

What did the programmer's ghost say?
BOOL!