Seneca College

Applied Arts & Technology

SCHOOL OF COMPUTER STUDIES

JAC444 **Submission date:** **Feb 08, 2023**

Workshop 2

Description:
The following workshop lets you practice basic java coding techniques, creating classes, methods, 2D array.

Task 1: (2D array Challenge)

Use a two-dimensional array to solve the following problem:

- A company has
  - Four salespeople (1 to 4) who sell five different products (1 to 5).
- Once a day, each salesperson passes in a slip for each type of product sold. Each slip contains the following:
  1. The salesperson number
  2. The product number
  3. The total dollar value of that product sold that day
- Thus, each salesperson passes in between 0 and 5 sales slips per day.
- Assume that the information from all the slips for last month is available.

Write an application that will read all this information for last month's sales and summarize the total sales by salesperson and by product.

- All totals should be stored in the two-dimensional array sales.
- After processing all the information for last month, display the results in tabular format, with each column representing a salesperson and each row representing a particular product. Cross-total each row to get the total sales of each product for last month.
- Cross-total each column to get the total sales by salesperson for last month.
- Your output should include these cross-totals to the right of the totaled rows and to the bottom of the totaled columns.

Hints:

- I would recommend creating two classes for this problem at the minimum
  - SalesSlip to manage the class behaviors
    - getPerson()
    - getProduct()
    - getValue()
- 2D array to represent products and salesperson

- For last month sales slips use random --- for example create 100 sale slip records as single dimension array and store.

Prod/Person	1	2	3	4	Total
1	496.33	510.58	760.16	752.16	2519.23
2	418.87	1.00	183.67	292.81	896.36
3	444.39	96.29	350.39	756.10	1647.17
4	314.42	555.31	670.22	748.43	2288.38
5	592.64	685.95	482.26	437.49	2198.34
Total	2266.65	1849.13	2446.70	2987.00	

This output is generated using random numbers no user input required.

Task 2: (Class Practice)

Write a class named **GroceryList** that represents a list of items to buy from the market, and another class named **GroceryItemOrder** that represents a request to purchase a particular item in a given quantity (example: 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 a grocery list will have no more than 10 items. A GroceryList object should have the following methods:

- 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:

- public GroceryItemOrder(String name, int quantity, double pricePerUnit) — *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 the given value*

Note: Students will be creating their own testing class for this task and your testing class should test all the functionalities. You are more than welcome to create a menu to do the grocery as well.

Workshop Header

/***
Workshop #
Course:*<subject type> - Semester*
Last Name:*<student last name>*
First Name:*<student first name>*
ID:*<student ID>*
Section:*<section name>*
This assignment represents my own work in accordance with Seneca Academic Policy.
Signature
Date:*<submission date>*
***/

Code Submission Criteria:

Please note that you should have:

- Appropriate indentation.
- Proper file structure
- Follow java naming convention
- Document all the classes properly using JavaDoc
- JavaDoc should be generated properly in the project
- Do Not have any debug/ useless code and/ or files in the assignment

Deliverables and Important Notes:

All these deliverables are supposed to be uploaded on the blackboard once done.

- You are supposed to create **video with voice** of your running solution. **(50%)**
  - Screen Video captured file should state your last name and id, like Ali_123456.mp4 (or whatever the extension of the file is)

    OR
  - Show your work during the lab
- A text file which will reflect on learning of your concepts in this workshop. **(20%)**
  - Should state your Full name and Id on the top of the file and save the file with your last name and id, like Ali_123456.txt
- JavaDocs must be used for proper documentation of each task. **(15%)**
- Submission of working code. **(15%)**
  - Make sure your follow the "**Code Submission Criteria"** mentioned above.
  - You should zip your whole working project to a file named after your Last Name followed by the first 3 digits of your student ID. For example, **Ali123.**zip.

- Your marks will be deducted according to what is missing from the above-mentioned submission details.
- Late submissions would result in additional 10% penalties for each day or part of it.
- Remember that you are encouraged to talk to each other, to the instructor, or to anyone else about any of the assignments, but the final solution may not be copied from any source.