CS 1341 - Lab 4 Programming Assignment

Arrays, Strings, more practice with multiple methods

Assignment Due Saturday, October 15th, 2022 at 6:00 AM Uploaded to Canvas

There is NO pre-lab for this assignment.

Density (15 POINTS)

Write a Java program called **`Density`** which does the following:

1. Uses a **`for`** loop which repeats 3 times. Within the loop,
   a. prompt the user to enter a mass in grams and store the mass in a **`double`** array called **`mass`**
   b. prompt the user to enter a volume in milliliters and store the volume in a **`double`** array called **`volume`**
2. Calls the method **`calculateDensity`** with the following header

```
public static void calculateDensity(double[] mass, double[] volume)
```

to calculate and display the density for each mass and volume (*Hint:* density = mass/volume). For completion, make sure that you include the units as shown in the sample output.

Sample output:

```
Enter the mass and volume for your 3 compounds.

Enter mass in grams: 3

Enter volume in milliliters: 23

Enter mass in grams: 4

Enter volume in milliliters: 56

Enter mass in grams: 65

Enter volume in milliliters: 1

The density for mass = 3.00 g and volume = 23.00 ml is: 0.13 g/ml

The density for mass = 4.00 g and volume = 56.00 ml is: 0.07 g/ml

The density for mass = 65.00 g and volume = 1.00 ml is: 65.00 g/ml
```

Seat Reservation (85 POINTS)

Write a Java program called **SeatReservation** which allows a user to reserve seats given a seating chart. The program will update and display the seating chart as the user selects seats, displays subtotals for the reservation, and finally displays the receipt including fees and taxes when the user is done with all the seat selection. You must use arrays, loops, and methods in your solution as described below.

main() method [20 points]

1. Create 4 arrays which contain the seats (**String[]** arrays) and their prices (**double[]** arrays) as follows:

   rowASeats

“1A”	“2A”	“3A”

   rowBSeats

“1B”	“2B”	“3B”

   rowAPrices

99.99	110.99	99.99

   rowBPrices

75.99	85.99	75.99

2. Create a **total** variable of type **double** which will keep track of the total price of the reservation so far
3. Call the **display()** method to display the seating chart and a welcome message
4. Ask the user how many seats they want to reserve and store the value in an **int** variable called **numSeats**
5. Then, in a for loop which will iterate **numSeats** times, do the following:
   a. Call the **getRow()** method to return the letter of the row from which the user wants to reserve a seat
   b. Check if the row is A or B, and call the **makeReservation()** method passing it the correct row and prices arrays. The method should return the price of the seat after the reservation is made which will be added to the **total** variable.
   c. Display the total so far.
6. Lastly, when the loop is done call the **printReceipt()** method to print the receipt

display() method [10 points]

1. Method header
   ```
   public static void display(String[] rowASeats, String[] rowBSeats,
   double[] rowAPrices, double[] rowBPrices)
   ```
2. Display the following welcome message. Call the methods **printRowSeat()** and **printRowPrices()** for rows A and B and their prices

```
Welcome to our event! Here's our seating chart with prices:
Seating Chart
===================================
|1A         2A         3A         |
|$99.99     $110.99    $99.99     |
-----------------------------------
|1B         2B         3B         |
|$75.99     $85.99     $75.99     |
===================================
```

printRowSeat() and *printRowPrices()* methods [10 points: 5 points each]

1. Method headers
   ```
   public static void printRowSeat(String[] rowSeats)
   public static void printRowPrices(double[] rowPrices)
   ```
2. Display the seats and prices, respectively, using a for loop in each method. Here's row A and row A prices as an example:

```
|1A         2A         3A         |
|$99.99     $110.99    $99.99     |
```

getRow() method [5 points]

1. Method header
   ```
   public static String getRow(Scanner scan)
   ```
2. Ask the user which row they want to reserve from and returns the letter representing the row

makeReservation() method [20 points]

1. Method header
   ```
   public static double makeReservation(String[] rowSeats, double[]
   prices, String row, Scanner scan)
   ```
2. Display row and prices for the selected row by calling ***printRowSeat()*** and ***printRowPrices()***
3. Asks the user which seat number they want and store the user's response in an **int** variable called **seatNum**
4. Display the user's selection (seatNum and row. Ex: 1A) and price

5. Call **updateSeatingChart()** method to display an “X” in place of the selected seat
6. Print the updated row with prices by calling ***printRowSeat()*** and ***printRowPrices()***
7. Return the price of the selected row

updateSeatingChart() method [10 points]

1. Method header
   public static void updateSeatingChart(String[] rowSeats, int seatNum)
2. Set the correct seat in the array to “X”

printReceipt() method [10 points]

1. Method header
   public static void printReceipt(int numSeats, double total)
2. Calculate fees ($14.99 per seat), taxes and total including fees and taxes
3. You can use 8.5% as the value for sales tax rate for this calculation.
4. Print the receipt as follows:

```
Thank you for reserving with us. Here's your receipt:
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Subtotal: $175.98
Fees: 2 x $14.99 = $29.98
Taxes: $16.99
=====================================================
Total: $222.95
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

HINTS and TIPS

- Since you created a Scanner object in the *main* method, you will pass the variable containing a reference to that Scanner to the *getRow* and *makeReservation* methods so you don’t need to create another Scanner in each of those methods.
- Although this may look like a complex problem when reading these instructions, it’s recommended that you build it incrementally. Start with the *main* method, then build and test one of the other methods before moving on to the next one. This will help you verify that each increment of your solution is working correctly before moving on to the next.
- When passing an array to the *makeReservation* and *updateSeatingChart* methods, you are passing an array by reference. This means that any changes you make to the array will be reflected when you return to the *main* method.

Here’s a complete run of the program.

Sample Output:

```
Welcome to our event! Here's our seating chart with prices:
Seating Chart
===================================
|1A         2A         3A         |
|$99.99     $110.99    $99.99     |
-----------------------------------
|1B         2B         3B         |
|$75.99     $85.99     $75.99     |
===================================
How many seats do you want to reserve?
2
Enter the row for the seat you want to reserve: (A/B)A
===================================
|1A         2A         3A         |
|$99.99     $110.99    $99.99     |
===================================
Which seat number do you want?
1
The seat you selected is: 1A
The price of the seat is: $99.99
Updated row chart:
===================================
|X          2A         3A         |
|$99.99     $110.99    $99.99     |
===================================
Your subtotal is: $99.99
-----
Enter the row for the seat you want to reserve: (A/B)B
===================================
|1B         2B         3B         |
|$75.99     $85.99     $75.99     |
===================================
Which seat number do you want?
1
The seat you selected is: 1B
The price of the seat is: $75.99



Updated row chart:
===================================
|X          2B         3B         |
|$75.99     $85.99     $75.99     |
===================================
Your subtotal is: $175.98
-----
Thank you for reserving with us. Here's your receipt:
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Subtotal: $175.98
Fees: 2 x $14.99 = $29.98
Taxes: $16.99
=====================================================
Total: $222.95
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

NOTES: Comment your program to explain your steps. Each program should compile without errors and should run to produce outputs described. The following points will be discounted if the related element is missing or incorrect:

- Output formatting monetary amounts (dollar sign and 2 places after the decimal point) [2 points each]
- Proper names for classes, variables, and methods [1 point each]
- No Comments [5 points]
- Program doesn't compile [5 points for each minor error up to 5 errors provided that after fixing the errors the program compiles. If the program does not compiler after the 5 errors are fixed, partial credit will be given not to exceed 50 points]
- Source code (java file) missing [20 points]
- Both java file and class file missing [100 points]
- Missing method where a method is required [5 points each]
- Missing loop where loop is required [5 points each]
- Missing array where array is required [10 points each]
- Logical errors in calculation [5 points each]

Submit the **Density.java**, **Density.class, SeatReservation.java and SeatReservation.class** files in one zipped file to Canvas by 6am 10/15/2022.