

COIT20245 Introduction to Programming (T1 2015)

Assessment item 2— Assignment 2

Due date: Thursday of Week 10

ASSESSMENT

Weighting: 30%

Length: NA

2

Objectives

This assessment item is designed to test your understanding of arrays/arrayLists, objects, classes, sorting and searching.

Assessment task

This assignment requires you to develop a java application program for such a scenario that a local department store manager wants to use it for calculating monthly commissions for N salespersons. N should be declared as a constant and set N=9 or N=10 for the purpose of test. The application also requires displaying, sorting and searching the sales data and corresponding commissions as shown in figures on next a few of pages.

Assume that the salespersons have just started the sales job in the local department store. Each salesperson earns only sales commission which is calculated by multiplying his/her monthly sales amount by the appropriate commission rate. The commission rate scheme has been set up by the store manager and is shown as below:

<u>Sales Amount</u>	<u>Commission Rate</u>
\$10,000 – \$25,000	5%
\$25,001–\$50,000	6%
\$50,001 – \$75,000	8%
\$75,001 – \$100,000	10%

From the above table, it can be seen that the commission rate is categorized into four levels according to the sales amount. The more the sales amount, the more the salesperson earns. The application should ask the user to enter a salesperson name and his/her monthly sales amount (In terms of the unit of \$1000. For example, If the entered number is 20, it means the sales amount is \$20,000). Assume that the sales amount is an integer number between 10 and 100 (this means the actual sales amount is between \$10,000 and \$100,000), according to the statistical results of many year sales data. An error message should be displayed and re-entering data is required, if the user enters the sales amount beyond this range. In addition, the input of a salesperson's name must be validated in such a way that a valid name is assumed only to contain English letters and a space between first name and surname. An example of an invalid input of name can be shown in the Figure 8 on page 6.

The data of salesperson name, sales amount and the calculated commission (unsorted) were stored in three arrays (The sample data in Figure 2 on next page contain 10 records. You can enter these sample data to test your program functionality). You must use parallel arrays or ArrayLists in this assignment to store salesperson name, sales amount and commissions. Your application should display and

execute a menu with the following options. A switch statement must be used to execute the following menu options.

1. Input & validate data
2. Display
3. Sort by name
4. Sort by sales
5. Search by name
6. Search by sales
7. Display statistics
8. Exit

The details for each option are described as below.

1. Input & validate data

This option reads each salesperson's name, and sales amount for N people from the keyboard and stores them in separate one-dimension arrays/arraylists. Figure 1 shows a sample input for a personal data. If the input of sales amount exceeds the ranges defined on the last page, then an appropriate message should be displayed and the user should be asked to re-enter a new value. In addition, the input of salesperson's name also must be validated as required on page 1.

2. Display

This option displays the salesperson names, sales amount and calculated commission that are stored in Arrays/ArrayLists for all sales persons, as shown in Figure 2.

3. Sort by name

When this option is selected, the names of sales person are sorted in ascending order and this option also displays sorted names with their sales amount and commission data, as shown in Figure 3. You can use any sorting algorithm which uses at least two while loops and one if statement. **A built-in sort should not be used.**

4. Sort by sales

This option sorts the sales amount in ascending order and displays the corresponding changes of salesperson names, and commissions, as shown in Figure 4.

5. Search by name

When the 'Search by name' option is executed and the user enters a salesperson's name regardless of the lower case or upper case in the name spelling, the sales amount, and commission for that person will be displayed as shown in Figure 5 (In this example, the salesperson's name – *david sydney* is entered). If the entered name doesn't match any existed one, it will show a warning message.

6. Search by sales

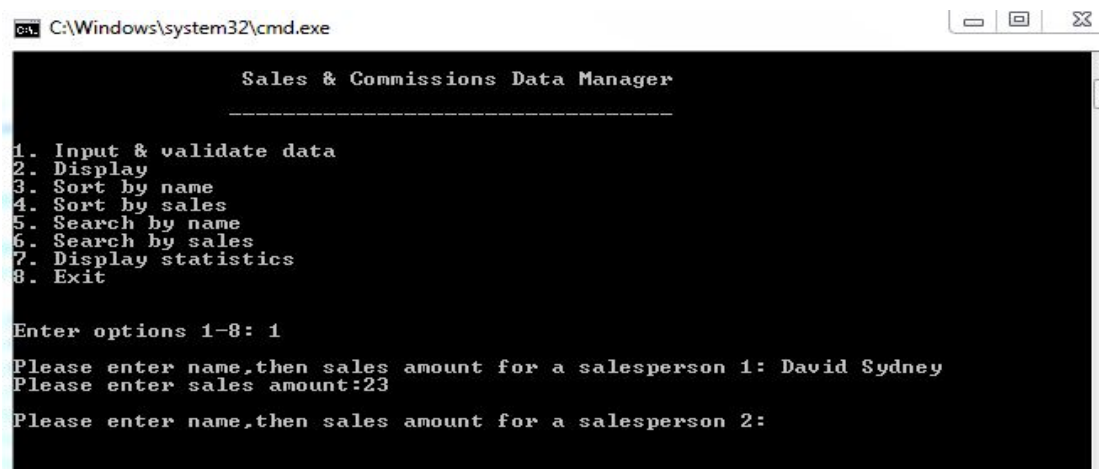
If the user selects this option and enters a particular sales amount, the running program will display the salesperson name and sales amount that are less than the particular amount. Figure 6 shows an example where the specific sales amount is \$50,000.

7. Display statistics

If the user selects this option, the program will calculate and display the lowest and highest value of sales amounts and corresponding salesperson names, as well the median sales amount, as shown in Figure 7.

8. Exit

Selecting 'Exit' option will exit the execution of the program.



```

C:\Windows\system32\cmd.exe

Sales & Commissions Data Manager
-----
1. Input & validate data
2. Display
3. Sort by name
4. Sort by sales
5. Search by name
6. Search by sales
7. Display statistics
8. Exit

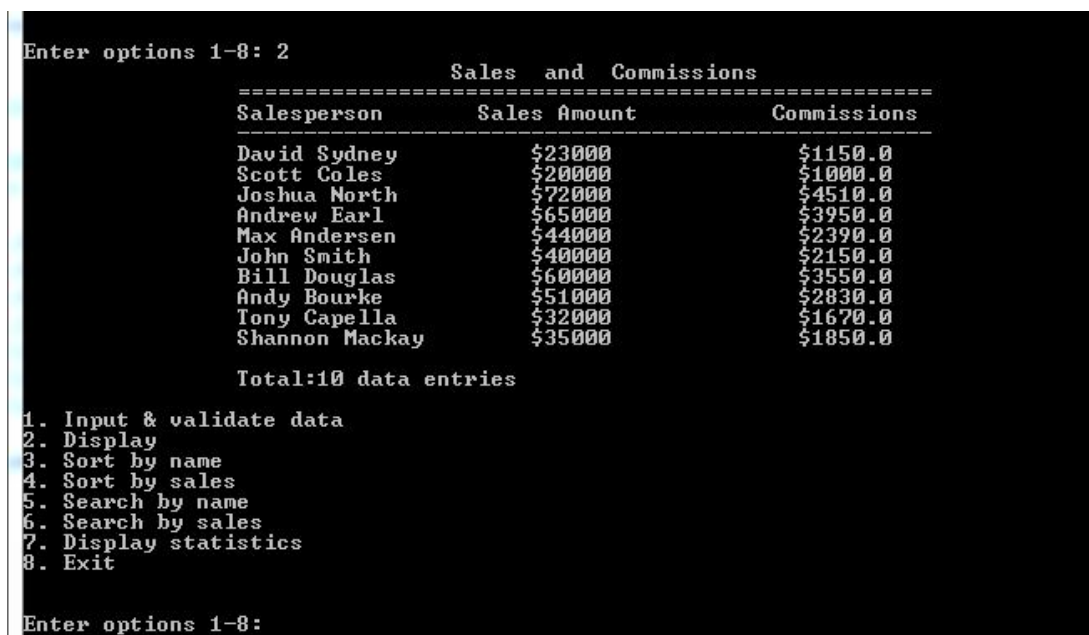
Enter options 1-8: 1

Please enter name, then sales amount for a salesperson 1: David Sydney
Please enter sales amount: 23

Please enter name, then sales amount for a salesperson 2:

```

Figure 1



```

Enter options 1-8: 2

Sales and Commissions
=====
Salesperson      Sales Amount      Commissions
-----
David Sydney     $23000            $1150.0
Scott Coles      $20000            $1000.0
Joshua North     $72000            $4510.0
Andrew Earl      $65000            $3950.0
Max Andersen     $44000            $2390.0
John Smith       $40000            $2150.0
Bill Douglas     $60000            $3550.0
Andy Bourke      $51000            $2830.0
Tony Capella     $32000            $1670.0
Shannon Mackay   $35000            $1850.0

Total:10 data entries

1. Input & validate data
2. Display
3. Sort by name
4. Sort by sales
5. Search by name
6. Search by sales
7. Display statistics
8. Exit

Enter options 1-8:

```

Figure 2

```
Enter options 1-8: 3

Sales and Commissions
=====
Salesperson      Sales Amount      Commissions
-----
Andrew Earl      $65000            $3950.0
Andy Bourke      $51000            $2830.0
Bill Douglas     $60000            $3550.0
David Sydney     $23000            $1150.0
John Smith       $40000            $2150.0
Joshua North     $72000            $4510.0
Max Andersen     $44000            $2390.0
Scott Coles      $20000            $1000.0
Shannon Mackay   $35000            $1850.0
Tony Capella     $32000            $1670.0

Total:10 data entries

1. Input & validate data
2. Display
3. Sort by name
4. Sort by sales
5. Search by name
6. Search by sales
7. Display statistics
8. Exit
```

Figure 3

```
Enter options 1-8: 4

Sales and Commissions
=====
Salesperson      Sales Amount      Commissions
-----
Scott Coles      $20000            $1000.0
David Sydney     $23000            $1150.0
Tony Capella     $32000            $1670.0
Shannon Mackay   $35000            $1850.0
John Smith       $40000            $2150.0
Max Andersen     $44000            $2390.0
Andy Bourke      $51000            $2830.0
Bill Douglas     $60000            $3550.0
Andrew Earl      $65000            $3950.0
Joshua North     $72000            $4510.0

Total:10 data entries

1. Input & validate data
2. Display
3. Sort by name
4. Sort by sales
5. Search by name
6. Search by sales
7. Display statistics
8. Exit
```

Figure 4

```

Enter options 1-8: 5
Please enter a sales person name:
david sydney

    David Sydney - sales amount: $23000 With commsisisions received:$1150.0
-----

1. Input & validate data
2. Display
3. Sort by name
4. Sort by sales
5. Search by name
6. Search by sales
7. Display statistics
8. Exit

```

Figure 5

```

Enter options 1-8: 6
Please enter a specific sales amount:
50
The following sales persons have less than $50000:

    Scott Coles $20000
    David Sydney $23000
    Tony Capella $32000
    Shannon Mackay $35000
    John Smith $40000
    Max Andersen $44000

1. Input & validate data
2. Display
3. Sort by name
4. Sort by sales
5. Search by name
6. Search by sales
7. Display statistics
8. Exit

```

Figure 6

```

Enter options 1-8: 7

                                A simple statistics
                                =====

    The sales person has lowest sales amount is Scott Coles, $20000.
    The sales person has highest sales amount is Joshua North, $72000.
    The median sales amount is $42000

1. Input & validate data
2. Display
3. Sort by name
4. Sort by sales
5. Search by name
6. Search by sales
7. Display statistics
8. Exit

Enter options 1-8:

```

Figure 7

```

Sales & Commissions Data Manager
-----
1. Input & validate data
2. Display
3. Sort by name
4. Sort by sales
5. Search by name
6. Search by sales
7. Display statistics
8. Exit

Enter options 1-8: 1

Please enter name, then sales amount for a salesperson 1: Jack123 London
You entered an invalid name.
Please input a valid name.
```

Figure 8

Program design

You may use any design that meets the specification. However, a good design will adhere to the following guidelines:

- be logically correct
- be easy to read and maintain
- be well-designed
- use UML class diagram
- use following methods and class

```
public class SalesDataManager
{
    public SalesDataManager( )           //constructor
    public void displayMenu()

    public void inputData()
    public void sortByName( )
    public void sortBySales( )
    public void searchByName( )
    public void searchBySales( )
    public void displayStatistics( )
    public boolean isValidName(String name)
    public double calculateCommission(int salesAmount)

    helper methods here if any

    public static main(String [ ] args)
    {
        .....
    }
}
```

Testing

Testing is important. You should:

- list the different types of test cases.
- display the results of each test case.

What to submit

You should submit online the following files:

- SalesDataManager.java (this file contains java source code for class SalesDataManager and main method)
- Report.docx (this file contains a brief report – maximum 2 pages that includes student name, student ID, course name, course code , test cases and UML class diagram for the class - SalesDataManager). The UML class diagram is a type of static structure diagram that describes the structure of a system by showing the class' data members and operations. A good example from the textbook is the UML class diagram for the Account class on page 494.

Assessment marking criteria

	Total Marks – 30	Marks Allocated
1	Design, logic, and testing -12	
	Code readability – appropriate use of comments, indentation and naming conventions	3
	Declaring and using variable	2
	Creating /declaring and using objects, methods and class	2
	Overall design and logic (structure, efficiency)	2
	Validity of testing as evidenced by submitted test results	2
	UML class diagram	1
2	Compilation and execution - 16	
	(0 mark – if your application doesn't execute)	
	Input and validation	3
	Display	2
	Sort by name	2
	Sort by sales	2
	Sorting algorithm implemented correctly	1
	Search by name option works with correct result	1.5
	Search by sales option works with correct result	1.5
	Display statistics works with correct result	2
	Exit option works	0.5
	Overall program execution	0.5
3	Report - 2	
	Presentation	2
4	Penalty	
	Submission of incorrect files including names and formats	up to -2
	Using techniques not covered in the course	up to -2
	Late submission	-1.5 mark per day