Assessment details for ALL students

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, methods, sorting and searching.

Assessment task

Write a java application that allows the user to read, validate, store, display, sort and search the account number and balance for N bank customers. N should be declared as a constant and it should be equal to the largest digit of your student id number (e.g. if your ID number is S0159226 then N should be equal to 9 and you can declare it as final int N=9;). The account number and balance must be stored in two separate single dimension arrays/arrayLists (index 0 for customer 1 and index N-1 for customer N). The minimum and maximum account numbers which can be stored are 999 and 9999. The minimum and maximum account balances which can be stored are $1 and $100000. The account number and balance must be entered from the keyboard and a validation for minimum and maximum values must be done.

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. Read, validate and store account number and balance for N customers
2. Display account number and balance for all customers
3. Display all account numbers which have $1 balance
4. Find and display the account balances which have below the average balance
5. Find and display the account number with the largest balance
6. Sort and display the account balances in descending order
7. Search and display the accounts which have balance equal to the given balance
8. Exit from application

1. Read, validate and store account number and balance for N customers
This option reads account number and balance for all customers from the keyboard and stores them in two separate single dimension arrays/arrayLists. If the account number is less than 999 and greater than 9999 then an appropriate message should be displayed and the user should be asked to enter a new account number. Similarly if the balance is less than $1 and greater than $100000 then an appropriate message should be displayed and the user should be asked to enter a balance.

2. Display account number and balance for all customers
This option displays the account number and balance stored in Arrays/ArrayLists for all customers.

3. Display all account numbers which have $1 balance
This option displays all account numbers stored in Array/ArrayList which have $1 balance.

4. Find and display the account balances which have below the average balance
This option finds and displays the account balances stored in Array/ArrayList which have balance below the average balance.

5. Find and display the account number with the largest balance
This option finds and displays the account number stored in Array/ArrayList which has largest balance. If there is more than one account with the same balance (largest balance) then it displays all of them.

6. Sort and display the account balances in descending order
This option sorts and displays account balances stored in Array/ArrayList for all customers in descending order. You must use the Selection Sort algorithm. **A built-in sort algorithm for sorting is not allowed in this assignment**.

7. Search and display the accounts which have balance equal to the given balance
This option searches Array/ArrayList and displays the accounts which have balance equal to the given balance. The application asks the user to enter a balance using the keyboard and searches for it. If the balance entered from the keyboard is found then the application displays account number and balance otherwise it displays an appropriate message. **A built-in search algorithm for searching can be used in this assignment**.

8. Exit from application
The application should display the message "Thank you" and your student id number and then exit from application.

The application should work in a loop to enable the user to read, validate and store account number and balance for N customers, display account number and balance for all customers, display all account numbers which have $1 balance, find and display the account balances which have below the average balance, find and display the account number with the largest balance, sort and display the account balances in descending order, search and display the customers which have balance equal to the given balance and exit from application.

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 activity diagram
- use appropriate classes, methods and fields

Testing

Testing is important. You should:

- List the different types of test cases.
- Display the results for each test case.

What to submit

You should submit online the following files. You can use any IDE (TextPad or NetBeans), however the program you submit must compile in TextPad. You will **get 0 mark for compilation & execution part, if your program doesn't compile in TextPad.**

- Account.java (this file contains java code for class Account)
- AccountApplicationt.java (this file contains java code for class AccountApplication).
- Report.docx (this file contains a brief report - maximum 2 pages that includes student name, student ID, course name, course code, UML activity diagram for menu option 3 (Display all account numbers which have $1 balance) and test results (screenshots/test cases with results to show that your application is correctly working)).

Assessment 2 marking criteria

	Total Marks – 30	**Marks Allocated**
1	**Design, logic and testing – 14**	
	Code readability – appropriate use of comments, indentation and naming conventions	3
	Declaring and using fields/variables and constants	2
	Creating/declaring and using objects, methods and classes	3
	Overall design and logic (structure, efficiency)	2
	Validity of testing as evidenced by submitted test results	2
	UML activity diagram for menu option 3	2
2	**Compilation and execution - 14**	
	(0 - if application doesn't run)	
	Welcome and exit messages	1
	Input data validation and message	1
	Read account number and balance for N customers	1
	Display account number and balance for all customers	1
	Display all account numbers which have $1 balance	2
	Find and display the account balances which have below the average balance	1
	Find and display the account number with the largest balance	1
	Sort and display the account balances in descending order (0 mark if selection sort is not used)	2
	Search and display the accounts which have balance equal to the given balance	2
	Overall program execution (user friendly, input/output and display)	2
3	**Report – 2**	
	Presentation	2