Introduction to Programming COIT 20245

Assessment item 1—Java Console Program

Due date:	**Week 7 T318** – Midnight, **Friday** 4 January 2018 Refer below for complete assessment item 1 requirements (**Assignment One**)	ASSESSMENT
Weighting:	20%	
Length:	N/A	1

Objectives

This assessment item relates to the course learning outcomes as in the Unit Profile.

Details

For this assignment, you are required to develop a **Java Console Program** to demonstrate you can use Java constructs including input/output via a command line, Java primitive and built-in data types, Java defined objects, selection and looping statements, methods, and various other Java commands. Your program must produce the correct results.
You are only allowed to use techniques which have been covered in the first six weeks of the course, you must use the **Scanner** object for input and **no advanced data structures like arrays** will be used. Instructions for this appear in the implementation section of this specification.

What to submit for this assignment

The Java source code:

- **NemoReefTours.java**

A report including a flow chart (UML activity diagram) to depict your validation loop for reading the number of passengers, how long it took to create, any problems encountered and screen shots of the output produced. (Use Alt-PrtScrn to capture just the console window and you can paste it into your Word document). You should test every possibility in the program.

- **ReportAss1.docx**

You will submit your files by the due date using the "**Assignment 1**" link on the Moodle unit website in the **Assessment Block** or in the relevant week.

Assignment specification

Nemo Reef Tours program

Nemo Reef Tours is company which takes passengers on a tour of the Great Barrier Reef, the tours depart at 8:00 am and returns around 5:00 pm. A smorgasbord lunch is also provided along with the opportunity to snorkel among the exotic fish and coral.

This program will allow staff at Nemo Reef Tours to compute the cost of a group of passengers to take the tour. There is a flat fee of eighty five dollars and fifty cents ($85.50) per person.

The management of Nemo Reef tours would like to encourage more passengers to take the tour, so it was decided there would be discounts for a larger number of passengers in the booking group. The discounts are as follows:

Charge per person $85.50 (as above).

One to two passengers: no discount.
From three to five passengers: 10% discount.
From six to ten passengers: 15% discount.
More than ten passengers: 20% discount.

The Nemo Reef Tours management has asked you to write a program to help employees determine the charges for booking a tour and to also produce some statistics

You are to write a **Java Console Application** (NemoReefTours.java) which will allow employees to enter the details of N booking names and the number of passengers for each booking. **N should be equal to the highest digit in your student ID**, use N=3 if your highest digit is less than three. For each booking the program will prompt for and accept the booking name and the number of passengers for the booking, it will then display the charge (see sample output below for formatting details).

For simplicity the bookings will be for the following day.

When all the bookings have been entered you need to report the maximum and minimum number of passengers per booking and the relevant booking name, the average number of passengers per booking and the total charges which have been collected.

The required **Java Console Application** should allow the user to:

1. For each of the N bookings: enter the ***Booking name***, and then enter the ***Number of passengers***. The program will output the charge for the booking. All dollar values will be formatted to two decimal places (see implementation below with help for doing this).
2. You must ensure the booking name is not blank so you must implement a validation loop so a booking name is entered. The number of passengers must be a positive number (non-zero) and you will also need to implement a validation loop so valid number of passengers are entered.

The program will number each booking in the input prompt.

3. When N bookings have been entered, you will output a heading for the statistics “Statistical information for Nemo Reef Tours”, the minimum and maximum number of passengers in the booking groups and the booking name with these minimums and maximums, and then what the average number of passengers per booking is (formatted to two decimal places) (see sample output below). Note: if more than one booking has an equal maximum or minimum passengers you just need to only output one booking name.

4. Display a welcome message at the beginning “Welcome to the Nemo Reef Tour Management System” and an end message e.g. “Thank you for using the Nemo Reef Tour Management System” and the final line “Program written by <your student ID>” (see sample output below).

The numeric literal values N, number of passengers for the different discounts and discounts ***must*** be represented as constants.

Implementation

A large number of students have never written a program before so this is a fairly simple assignment which can be written in the main method of your class. Follow the steps outlined here and build your program up in a step by step fashion and always compile your program at each stage so you are always working on error free code.

Start by creating your NemoReefTours class which will contain just the main method, COMPILE! (Fix any errors and repeat)

Implement the welcome message, COMPILE and RUN!

Declare your Scanner object(s), COMPILE!

> Note: In order to combat the problem of the Scanner objects reading both textual and numeric data a good way to counter this is to declare two Scanner objects, one for reading text and another for reading numbers, or you can clear the buffer after the int read using `nextLine()`

Create a loop to loop N Times, COMPILE! (use N = 3 for development)

Declare variables to hold the booking name and the number of passengers (String and int), COMPILE!

Within the loop: prompt and read the booking name, COMPILE and RUN!

Add the prompt and read for the number of passengers, COMPILE and RUN!

Calculate the tour charges using the pricing structure above, COMPILE, RUN and TEST until this is correct.

Output the description of the transaction (see sample output below)
Output all dollar values and average to two decimal places:
USE: `System.out.printf("%.2f", charge);`

Add the validation loops for reading the data (you can do this last if you like)

Use if statements to determine if the number of passengers is maximum or minimum, (you will have to think about this). Output the minimum and maximums after the loop, COMPILE and RUN until you have this correct. You may want to set your original max and min variables to very small and very large numbers using `Integer.MIN_VALUE` and `Integer.MAX_VALUE.`

You will need to add up the number of passengers as you go so you can calculate the average.

After the loop you will output the statistics which you have gathered in the loop. You should have the maximum and minimum number of passengers and the corresponding booking names and the total amount of charges collected.

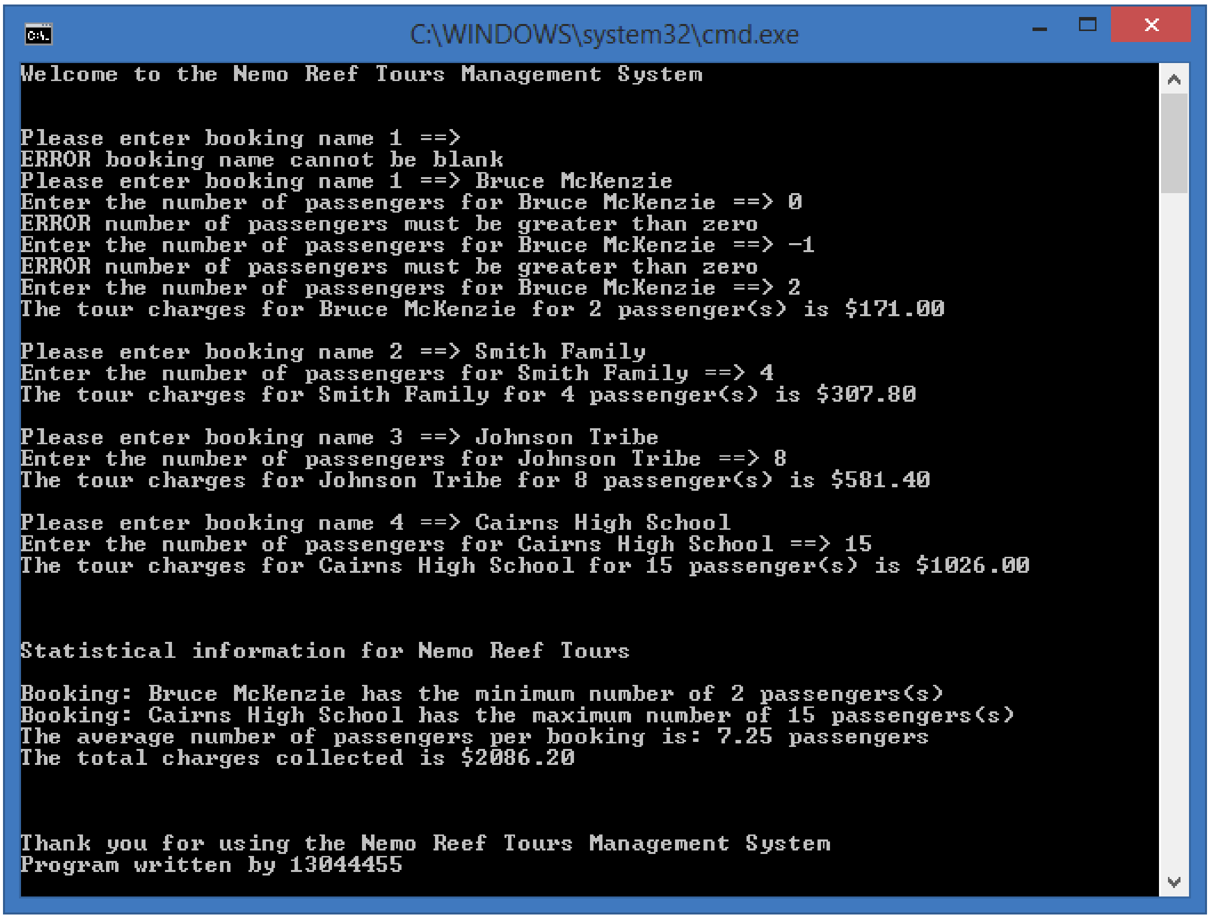
Calculate the average number of passengers per booking.

Output the statistics as indicated in the screen shot below.

Finally print the end message.

Your program should be well laid out, commented and uses appropriate and consistent names (camel notation) for all variables and objects.
For this assignment you will not worry about checking data types.
Refer to a Java reference textbook and the unit and lecture material (available on the course WEB site) for further information about the Java programming topics required to complete this assignment.
Check the marking guide (last page) to ensure you have completed every task. You need to match the output as shown below.

Typical screen shot of the program executing is as follows:



```
Welcome to the Nemo Reef Tours Management System


Please enter booking name 1 ==>
ERROR booking name cannot be blank
Please enter booking name 1 ==> Bruce McKenzie
Enter the number of passengers for Bruce McKenzie ==> 0
ERROR number of passengers must be greater than zero
Enter the number of passengers for Bruce McKenzie ==> -1
ERROR number of passengers must be greater than zero
Enter the number of passengers for Bruce McKenzie ==> 2
The tour charges for Bruce McKenzie for 2 passenger(s) is $171.00

Please enter booking name 2 ==> Smith Family
Enter the number of passengers for Smith Family ==> 4
The tour charges for Smith Family for 4 passenger(s) is $307.80

Please enter booking name 3 ==> Johnson Tribe
Enter the number of passengers for Johnson Tribe ==> 8
The tour charges for Johnson Tribe for 8 passenger(s) is $581.40

Please enter booking name 4 ==> Cairns High School
Enter the number of passengers for Cairns High School ==> 15
The tour charges for Cairns High School for 15 passenger(s) is $1026.00



Statistical information for Nemo Reef Tours

Booking: Bruce McKenzie has the minimum number of 2 passengers(s)
Booking: Cairns High School has the maximum number of 15 passengers(s)
The average number of passengers per booking is: 7.25 passengers
The total charges collected is $2086.20



Thank you for using the Nemo Reef Tours Management System
Program written by 13044455
```

Good luck --- Bruce McKenzie unit coordinator term 3 2018

b.mckenzie@cqu.edu.au

Marking Scheme

	Marks allocated
Total number of marks – 20	
Variables, constants and types	
Constant N is used correctly and is largest digit of student ID	0.5
Constants are used for all numeric literals (no hard coded values)	0.5
Variables have meaningful names and use camel notation	0.5
Variables are the correct type	0.5
Code in general	
Code is indented and aligned correctly	0.5
Code has header comment which includes student name, student ID, date, file name and purpose of the class	0.5
Code is fully commented including all variables	0.5
Input	
Numbers and strings are read correctly	0.5
Validation loop for booking name is correct	1
Validation loop for number of passengers is correct	1
Processing	
if statements are correct	0.5
looping is correct i.e iterates N times	0.5
Output	
Output is formatted correctly (resembles sample output)	2
Correct charge per booking is correct	2
Maximum passengers and booking name is correct	1
Minimum passengers and booking name is correct	1
Average passengers per booking is correct	1
Total charges is correct	0.5
Average and dollar values formatted to two decimal places	1
Welcome and exit message (with student ID) are displayed	0.5
General	
Correct files submitted including types and names	1
Only techniques covered during weeks 1-6 are used	1
Report	
Flow chart is correct	1
Screen shot(s) of testing	0.5
Report presentation and comments including how long it took and any problems encountered	0.5