COIT20245 Introduction to Programming

Assessment item 1— Assignment 1 Specification

Due date:	Thursday of Week 6	**ASSESSMENT**
Weighting:	20%	
Length:	NA	**1**

Objectives

This assessment item is designed to test your understanding of variables, constants, types, operators, standard console input/output, loops, if statements, classes, objects and methods.

Assessment task

Write a Java application that calculates and displays rentals of hiring car for N customers who hire cars from a local car rental company based in Rockhampton, Australia. N should be declared as a constant and should be equal to the largest digit of your student id number (e.g. if your student id is s0705544 then N should be equal to 7). Assume N=4, if the largest digit of your student id number is less than 4. The standard charges of car rental from this company are shown below in Table 1.

Table 1. The standard charges for car rentals

Days of rental	Charge
1 – 5 days	$25.0/per day (base rate)
6 – 10 days	$22.5/per day
More than 10 days	$21.0/per day

The application should ask the user to enter the customer name, the number of days for renting a car, and yes or no to indicate whether the customer receives a special offer or not. The special offer will give customers a 10% discount of the rental no matter how many days they hire a car. Assume that the number of days to be entered is an integer type with the range between 1 and 365. An error message should be issued if a user enters a value beyond this range and the re-entering is required. After entering the above three data items, the program makes a calculation of rental and displays the result as shown in Figure 1. The program should be kept running with entering next set of input data. After all N sets of input data are entered from the keyboard, if the calculation of your program is successfully finished, the execution of your program will also display the information that includes the customer spending most rental, the customer spending least rental as well as a simple statistics bar chart to display how many rents with the number of days are less than 7 days and greater than/equal 7 days in two lines respectively.(Based on historical statistics, most customers hire a car for a period of 7 days). These requirements and the result also can be shown in Figure 2 on next page. Your program should display a similar screen image when it runs. The application should be user-friendly by displaying appropriate welcome, exit and error message. Please note: (1)For simplicity, ignore some complicated factors such as car type, fuel usage, running kilometres in the real-world application. This is a simplified computation model for our purpose of programming exercise. (2)A special offer can be represented by a boolean variable with the value of true/false, or an integer variable with the

value of 1 and 0, or a String variable with the value of *yes* or *no* (you can use String class built-in method `equal( )` to compare two strings); and (3) If the value of N is re-set, your program should be run as normal without changing any source code except setting N with a different value.

The application can be implemented in a few of different ways. One of implementations is to use the following classes and methods:

```
public class CarRental
{
   //declare data members
   public CarRental( )    //constructor
   public void setName(String n)
   public String getName()
   public void setDays(int d)
   public int getDays()
   ……...
   public double calulateRental()
   public void displayInfo()
}

public class CarRentalTest
{
    public static main(String [ ] args)
    {
         //declare data members
         // create Scanner object, create a CarRental Object
         // use a loop to read data, call methods
         …….
    }
}
```

Alternatively, the another implementation can be illustrated as below:

```
public class CarRental
{
   //declare data members
   public CarRental( )    //constructor
   public void inputData()// using Scanner, using loops

   public double calulateRental()
   public void displayInfo()
}

public class CarRentalTest
{
    public static main(String [ ] args)
    {
        //simply create a CarRental object
        //call relevant methods
    }
}
```

You can implement this application in either way as above (or other similar way), provided your program works fine and produces the expected result that meets the specified requirements.

Example for N=7 (Note that a loop should be used for reading the input and displaying the output)

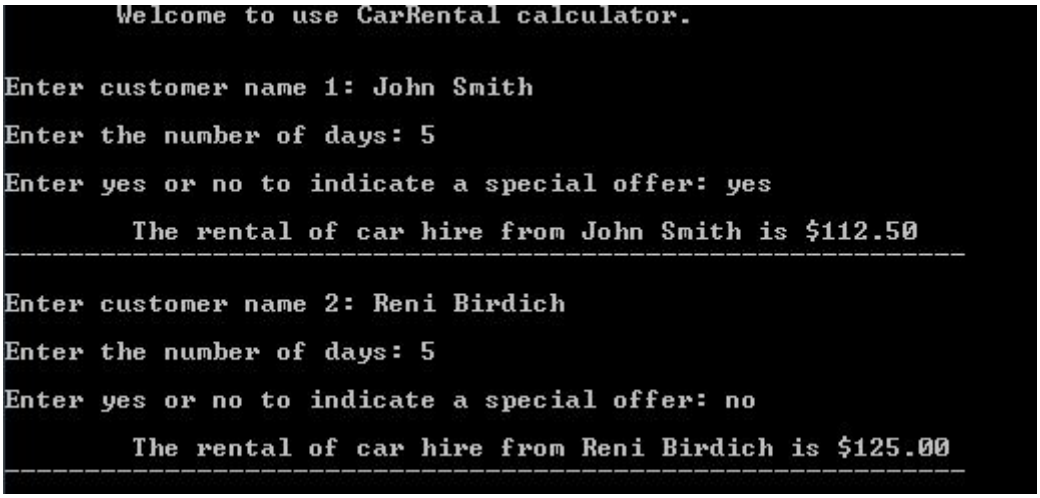

Figure 1. Sample input and running result(only display 2 inputs, ignore others)

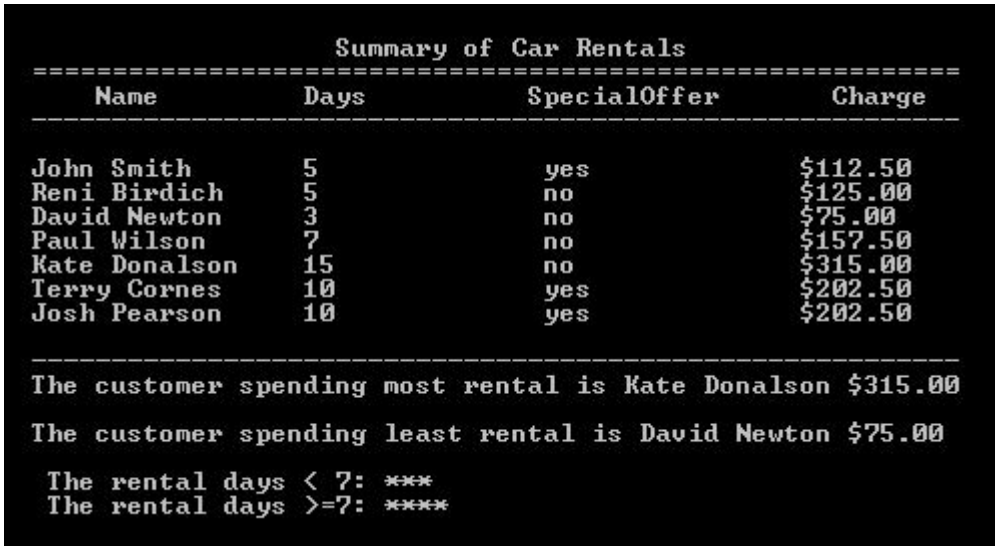

Figure 2. Sample running result of the program

What to submit

You should submit online the following files:

- CarRental.java (this file contains java source code for class CarRental)
- CarRentalTest.java (this file contains java source code for class CarRentalTest)
- Report.docx (this file contains a brief report – maximum 2 pages that includes student name, student ID, course name, course code and test cases).

Assessment marking criteria

	Total Marks – 20	**Marks Allocated**
1	**Variables, constants and types**	
	Declaring and using variables and constants	1
2	**Objects and classes**	
	Creating/declaring and using constructor, objects and classes	2
3	**Loops**	
	Using loops and conditions	1
4	**If statements**	
	Using if statements and conditions	1
5	**Methods**	
	Declaring and using main method	1
	Declaring and using other methods	2
6	**Inputs and Outputs**	
	Reading inputs and displaying outputs	1
	Validation of input – the number of days	0.5
	Rental calculation for each customer	1
	Summary of Rental as shown in Fig. 2	2.5
7	**Overall logic and program**	
	Program logic	1
	Spacing and Indentation conventions	1
	Naming conventions	1
	Comments in program	1
	User-friendly (welcome, exit and error messages)	1
8	**Testing**	
	Test cases	1
9	**Report**	
	Presentation	1
10	**Penalty**	
	For submission of incorrect files up to -2 marks	
	For using things not covered in Week 1-6 of this course (such as arrays) up to -3 marks	
	For late submission is -5% each day	