ASSESSMENT BRIEF

COURSE: Bachelor of Information Technology	
Unit:	Object Oriented Design and Programming
Unit Code:	OODP101
Type of Assessment:	Assessment 3 – Solution to programming problem by group of 3-4 students
Length/Duration:	20 Hours
Unit Learning Outcomes addressed:	Upon successful completion of this unit students should be able to: 1. Demonstrate basic knowledge of object-oriented programming concepts and programming problems 2. Analyse and dissect simple design and programming problem 3. Implement a well-designed modularized solution to small programming problems 4. Develop and/or implement testing schedules
Submission Date:	Week 8
Assessment Task:	A group of 3-4 students will work together to provide a quality solution to programming problem using JAVA programming language,
Total Mark:	20 marks
Weighting:	Converted to 20% of the unit total marks

Students are advised that **submission of an Assessment Task past the due date without a formally signed approved Assignment Extension Form** (Kent Website *MyKent Student Link> FORM – Assignment Extension Application Form – Student Login Required)* **or previously approved application for other extenuating circumstances impacting course of study, incurs a 5% penalty per calendar day**, calculated by deduction from the total mark.
For example. An Assessment Task marked out of 40 will incur a 2 mark penalty for each calendar day.

More information, please refer to *(*Kent Website *MyKent Student Link> POLICY – Assessment Policy & Procedures – Student Login Required)*

Kent Institute Australia Pty. Ltd.
ABN 49 003 577 302 CRICOS Code: 00161E RTO Code: 90458
TEQSA Provider Number: PRV12051

ASSESSMENT DESCRIPTION:

Your task is to design, develop and test a small application which will allow an user to estimate and compare the cost of the different ride sharing companies and find the most expensive and cheapest from them.

Task 1- Design

This stage requires you to prepare documentation that describes the function of the program and how it is to be tested. There is no coding or code testing involved in this stage.

Requirements:

1) Read through *Task 2: Program Development* to obtain details of the requirements of this program.
2) Write pseudocode that describes how the program will operate.
    a. All program requirements must be included, even if you do not end up including all these requirements in your program code.
    b. The pseudocode must be structured logically so that the program would function correctly.
3) Prepare and document test cases that can be used to check that the program works correctly once it has been coded. You do NOT need to actually run the test cases in this stage; this will occur in *Task 3: Testing*.
    a. Test cases should be documented using a template which is week 6 lecture and tutorial. You may include extra information if you wish. At this stage, the Actual Result column will be left blank. Two test cases per group member are required to gain full marks in this task.

Task 2: Program Development

Using the Design Documentation to assist you, develop a Java program that allows the user to enter their name, approximate kilometres, date and time of travel. Program will estimate the charges for travel according to different ride sharing companies and then compare the charges to give an idea to user about cheapest and expensive ride sharing company.

All requirements require that you follow coding conventions, such as proper layout of code, using naming conventions and writing meaningful comments throughout your program.

Requirement 1:

Display a welcome message when the program starts

- The welcome message should have a row of "*" at the top and the bottom, just long enough to extend over the text. *Hint: Use a loop for this.*
- The first line of the message should read "WELCOME TO RIDE-SHARING CHARGES ESTIMATOR AND COMPARISON SYSTEM"
- The second line of the message should be blank.
- The third line should read "Developed by" followed by your names and a comma, then "student ID", then your student ids of all group members.
- The fourth line should display "OODP101 Object Oriented Design and Programming"
- The fifth line should display the current date and time of system. You are expected to do a research to complete this task.
- The sixth line should be blank, and the seventh line should be another row of "*"

```
**************************************************:

Developed by Student Names, Student IDs
OODP101 Object Oriented Design and Programming
Mon Apr 12 11:17:25 AEST 2021

**************************************************:
```

Requirement 2

Provide a menu from which the user can select to Enter User Details, Display Charges Under Company 1, Display Charges Under Company 2, Display Charges Under Company 3, Show Report, or Exit System. Company names can be chosen by students from the numerous rides sharing companies available in Australia. This menu should be repeated each time after the user has chosen and completed an option until the user chooses to Exit. The user selects an option by entering the number next to it. If an invalid number is selected, the user is advised to make another selection.

```
**************************************************************************

Developed by Student Names, Student IDs
OODP101 Object Oriented Design and Programming
Mon Apr 12 15:05:10 AEST 2021

**************************************************************************
1. Enter Usage Details
2. Display charges under company 1
3. Display charges under company 2
4. Display cost under company 3
5. Show Report
6. Exit
7
Error!!!! You have entered invalid value!!! Please enter valid value
1. Enter Usage Details
2. Display charges under company 1
3. Display charges under company 2
4. Display cost under company 3
5. Show Report
6. Exit
1
Enter your name
```

Requirement 3

When the user selects the Enter User Details option, ask user to enter name, approximate KM, day, month and time of travel using 24 hour clock notation (for example:- 9.00 for 9 o'clock in morning, 13.00 for 1 o'clock in afternoon, 14.30 for 2:30 in afternoon) or Return to Main Menu.

***Note**:* We will use these to find out if it is falling in weekend category or not and We will use 2021 as a default year.

All numeric values entered must be positive otherwise program should ask user to enter the value again. After taking inputs, program should show the main menu.

```
1. Enter Usage Details
2. Display charges under company 1
3. Display charges under company 2
4. Display cost under company 3
5. Show Report
6. Exit
1
Enter your name
Danielle
Enter your approximate kilometers of travel
23
Month of travel
6
On which date of this month , you wish to travel!!
15
Now enter the time of travel using 24 hour clock just like 9.00 for 9am, 13.00 for 1 pm, 14.30 for 2:30pm etc
17.30
```

Requirement 4

When the user selects the Display Charges Under Company 1 option, the program should calculate the charges and display a summary of the user details which have been entered, and their charges under Company 1. While calculating, program should check if the date and time falls under peak time category or weekend category and display a message to user confirming if their travel time is falling any category.

Any time between 7.00 -9.00 am and 16.00-18.00 pm lies under peak time category and final charges should add peak time charges into it.

Saturday and Sundays falls under weekend category. Use the date input value to find out the day on which user wish to travel.

Visit these pages for help:

LocalDate (Java Platform SE 8) (oracle.com)

DayOfWeek getValue() method in Java with Examples - GeeksforGeeks

Actual ride sharing companies use many factors to calculate the charges but to keep our system simpler, only few factors will be considered.

The cost structure for Company 1 is listed in the following table. Once the charges have been displayed, it's total value should be saved into an array and the program should return to the Main Menu.

Items	Charges
Base Charges	$5.50
$/Km	$0.75
Peak Time Surcharges per trip	$2.50
Weekend Surcharges per trip	$3.00

```
Enter your choice using numbers
1. Enter Usage Details
2. Display charges under company 1
3. Display charges under company 2
4. Display cost under company 3
5. Show Report
6. Exit
2
********************************************************************************
Your travel details:- Day of the Week on 15 of the month - JUNE is TUESDAY
Your day of travel does not falls under weekend category

Time falls in peaktime category

So charges will be applied accordingly

The final charges under company 1 is: 25.25
********************************************************************************
1. Enter Usage Details
2. Display charges under company 1
3. Display charges under company 2
4. Display cost under company 3
5. Show Report
6. Exit
```

Note:-It is just a general output and you need to be creative while displaying the final cost to user. Do a bit of research on formatting output.

Requirement 5

When the user selects the Display Charges under Company 2 option, the program should do the same as in Step 4, but using Company 2's cost structure instead, which is listed in the following table, and then return to the Main Menu.

Items	**Charges**
Base Charges	$4.50
$/Km	$0.85
Peak Time Surcharges per trip	$2.00
Weekend Surcharges per trip	$2.50

Requirement 6

When the user selects the Display Charges Under Company 3 option, the program should do the same as in requirement no 4, but using cost structure that will be developed by you and then return to the Main Menu.

Items	**Charges**
Base Charges	
$/Km	
Peak Time Surcharges per trip	
Weekend Surcharges per trip	

Requirement 7

When the user selects show report option then program should show the name of cheapest and expensive company and return to main menu.

Requirement 8

When the user selects Exit System, the value of all variables related to the usage should all be reset to 0. A message reporting this, should be displayed, and then program should be terminated with a goodbye message.

Requirement 9

Modularize the code, correctly using method calls and passing data between methods as parameters.

Task 3: Testing

After finishing the development, test your program with the help of test cases developed task 1. In this task, you will be giving the actual output of your program. Make sure you provide screenshots in your report of all actual outcome of all test cases. You don't need to rewrite the test cases in this task but you definitely need to provide the proper numbers so your teacher can identify the relevant test cases from your screenshots.

ASSESSMENT SUBMISSION:

This submission will have one word/pdf and one java file.
This assignment should be submitted online in Moodle **.**

The assignment MUST be submitted electronically in Microsoft Word format. Other formats may not be readable by markers. Please be aware that any assessments submitted in other formats will be considered LATE and will lose marks until it is presented in Word.

For assistance please speak to our Academic Learning Skills Coordinators, in Sydney (als_syd@kent.edu.au) or in Melbourne (als_mel@kent.edu.au). They can help you with understanding the task, draft checking, structure, referencing and other assignment-related matters.

GENERAL NOTES FOR ASSESSMENT TASKS

Content for Assessment Task papers should incorporate a formal introduction, main points and conclusion.

Appropriate academic writing and referencing are inevitable academic skills that you must develop and demonstrate in work being presented for assessment. The content of high quality work presented by a student must be fully referenced within-text citations and a Reference List at the end. Kent strongly recommends you refer to the Academic Learning Support Workshop materials available on the Kent Learning Management System (Moodle). For details please click the link http://moodle.kent.edu.au/kentmoodle/mod/folder/view.php?id=3606 and download the file titled "Harvard Referencing Workbook". This Moodle Site is the location for Workbooks and information that are presented to Kent Students in the ALS Workshops conducted at the beginning of each Trimester.

Kent recommends a minimum of **FIVE (5)** references in work being presented for assessment. Unless otherwise specifically instructed by your Lecturer or as detailed in the Unit Outline for the specific Assessment Task, any paper with less than five (5) references may be deemed not meeting a satisfactory standard and possibly be failed.

Content in Assessment tasks that includes sources that are not properly referenced according to the "*Harvard Referencing Workbook*" will be penalised.

Marks will be deducted for failure to adhere to the word count if this is specifically stated for the Assessment Task in the Unit Outline. As a general rule there is an allowable discretionary variance to the word count in that it is generally accepted that a student may go over or under by 10% than the stated length.

General Notes for Referencing

References are assessed for their quality. Students should draw on quality academic sources, such as books, chapters from edited books, journals etc. The textbook for the Unit of study can be used as a reference, but not the Lecturer Notes. The Assessor will want to see evidence that a student is capable of conducting their own research. Also, in order to help Assessors determine a student's understanding of the work they cite, all in-text references (not just direct quotes) must include the specific page number(s) if shown in the original. Before preparing your Assessment Task or own contribution, please review this 'YouTube' video (Avoiding Plagiarism through Referencing) by clicking on the following link: link: http://moodle.kent.edu.au/kentmoodle/mod/folder/view.php?id=3606

A search for peer-reviewed journal articles may also assist students. These type of journal articles can be located in the online journal databases and can be accessed from the Kent Library homepage. Wikipedia, online dictionaries and online encyclopaedias are acceptable as a starting point to gain knowledge about a topic, but should not be over-used – these should constitute no more than 10% of your total list of references/sources. Additional information and literature can be used where these are produced by legitimate sources, such as government departments, research institutes such as the National Health and Medical Research Council (NHMRC), or international organisations such as the World Health Organisation (WHO). Legitimate organisations and government departments produce peer reviewed reports and articles and are therefore very useful and mostly very current. The content of the following link explains why it is not acceptable to use non-peer reviewed websites (Why can't I just Google?): https://www.youtube.com/watch?v=N39mnu1Pkgw
(Thank you to La Trobe University for access to this video).

Marking Guide (Rubric): It is just a draft rubric. It will be finalized by week 7.

Task-1 Design Pseudocode	Not attempted ***0points***		All requirements from task 2 have not been covered in English like representation of code OR No proper notation has been used as per lecture slides. ***1points***	All requirements from task 2 have been covered in English like representation of code and proper notation has been used as per lecture slides. ***2points***
Task 1- Test Cases	Not attempted ***0points***	No Two per person AND Not All test cases have tester name, proper description, input, expected output. ***1points***	Two test cases per member But Not All test cases have tester name, proper description, input, expected output. ***1.5points***	Two Test cases per member. AND All test cases have tester name, proper description, input, expected output ***2points***

Task 2- Requirement 1- Welcome Message	Not attempted ***0points***	Welcome message does not have three of the following: - 1. Line of stars before and after the message displayed using loop 2. Welcome to system 3. All student details 4. Current date and time ***0.5points***	Welcome message does not have two of the following: - 1. Line of stars before and after the message displayed using loop 2. Welcome to system 3. All student details 4. Current date and time ***1points***	Welcome message does not have 1 of the following: - 1. Line of stars before and after the message displayed using loop 2. Welcome to system 3. All student details 4. Current date and time ***1.5points***	Welcome message has all of the following: - 1. Line of stars before and after the message displayed using loop 2. Welcome to system 3. All student details 4. Current date and time ***2points***
Task 2- Requirement 2-Main Menu	Not attempted ***0points***	Main Menu is displayed with all options. AND Invalid inputs are not handled properly AND proper loop has not been used to display the menu again and again until user chooses to exit. ***0.5points***	Main Menu is displayed with all options. AND Invalid inputs are not handled properly OR proper loop has not been used to display the menu again and again until user chooses to exit. ***1points***	Main Menu is displayed with all options. AND Invalid inputs are handled properly AND proper loop has been used to display the menu again and again until user chooses to exit. ***1.5points***	
Task 2- Requirement 3- Enter Usage Details	Not attempted ***0points***	User is not asked to enter name, kms, month, day and time of travel OR invalid inputs like negative values are not handled properly. ***0.75points***	User is asked to enter name, kms, month, day and time of travel AND invalid inputs like negative values are handled properly. ***1.5points***		
Task 2- Requirement 4- Charges under company 1	Not attempted ***0points***	Total cost is not calculated using given: - Base Price $/km Weekend charges and peak charges. Proper coding is done to find out if user	Total cost is not calculated using given: - Base Price $/km Weekend charges and peak charges. Proper coding is done to find out if user	Total cost is calculated using given: - Base Price $/km Weekend charges and peak charges. Proper coding is done to find out if user	

		time of travel falls under weekend or peak time category. AND Cost is not saved in array and OR not displayed to user. ***0.5points***	time of travel falls under weekend or peak time category. OR Cost is not saved in array and OR not displayed to user. ***1points***	time of travel falls under weekend or peak time category. Cost is saved in array and displayed to user. ***1.5points***
Task 2- Requirement 5- Charges under company 2	Not attempted ***0points***	Total cost is not calculated using given: - Base Price $/km Weekend charges and peak charges. Proper coding is done to find out if user time of travel falls under weekend or peak time category. AND Cost is not saved in array and OR not displayed to user. ***0.5points***	Total cost is not calculated using given: - Base Price $/km Weekend charges and peak charges. Proper coding is done to find out if user time of travel falls under weekend or peak time category. OR Cost is not saved in array and OR not displayed to user. ***1points***	Total cost is calculated using given: - Base Price $/km Weekend charges and peak charges. Proper coding is done to find out if user time of travel falls under weekend or peak time category. Cost is saved in array and displayed to user. ***1.5points***
Task 2- Requirement 6- Charges under company 3	Not attempted ***0points***	Total cost is not calculated using own: - Base Price $/km Weekend charges and peak charges. Proper coding is done to find out if user time of travel falls under weekend or peak time category. AND Cost is not saved in array and OR not displayed to user. ***0.5points***	Total cost is not calculated using own: - Base Price $/km Weekend charges and peak charges. Proper coding is done to find out if user time of travel falls under weekend or peak time category. OR Cost is not saved in array and OR not displayed to user. ***1points***	Total cost is calculated using own: - Base Price $/km Weekend charges and peak charges. Proper coding is done to find out if user time of travel falls under weekend or peak time category. Cost is saved in array and displayed to user. ***1.5points***

Task 2- Requirement 7- Show Report	Not attempted ***0points***	Not reading from array OR not displaying both cheapest and expensive provider ***0.75points***	Reading elements from array and displaying both cheapest and expensive provider ***1.5points***
Task 2- Requirement 8- Exit	Not attempted ***0points***	Program exits but not resetting all values to zero OR There is a no proper code to set all values to zero. ***0.5points***	Program exits after resetting all values to zero. There is a proper code to set all values to zero. ***1points***
Task 2- Requirement 9- Modularizatio n	Not attempted ***0points***	Four modules have been created but all of them do not demonstrates three types of modules discussed in class. ***1points***	Four modules have been created AND all of them demonstrates three types of modules discussed in class. ***2points***
Task 2- Requirement 10- Proper coding standards and comments in code	Not attempted ***0points***	Comments are not enough OR no proper indentation and layout. ***0.5points***	Proper comments to explain each block of code, proper indentation, proper code layout. ***1points***
Task 3- Testing	Not attempted ***0points***	Only screenshots are provided but cannot be identified clearly for which test cases they are given OR all screenshots are not provided. ***0.5points***	Proper screenshots for all test cases and can be identified clearly. ***1points***