

## FIT1051 Assessment Four

**Submission deadline:** Monday 28<sup>th</sup> October 11:55pm AEST via Moodle

**Weight:** This assessment is worth 38% of the unit total. Submission of your code is worth 50% of this total, and your interview component (held online in the following week) is worth 50%.

**Late Penalty:** 5% mark deduction per day

**Instructions:** Below are the coding tasks that you need to complete for Assessment 4. Your work and your submission should be independent. Given the flexible nature of this assignment regarding classes etc., there is no ZIP file to download. Please design your solution with the appropriate .java files, and ZIP up the project and submit to Moodle upon completion.

**Academic Integrity:** Please be reminded of the academic integrity standards that are expected of you at Monash, which were mentioned in Week 1. You should code alone and ask the unit staff for help if needed. Do not post your code in public forums or send your code to anyone. Do not copy/paste code from other sources and present it as your own — this includes use of generative AI tools. Breaching these academic integrity requirements can incur serious penalties.

You are very much encouraged to undertake your own research to complete this assignment. However, if your code uses techniques that have not been covered in this course and you cannot explain them adequately in your interview, you will be at risk of failing the assignment.

**Scenario:** You are required to develop a system for a taxi company called *Zoomer*. Customers phone a call centre and can book one of three types of vehicle: *ZoomerLite*, *ZoomerStandard*, and *ZoomerPro*.

- *ZoomerLite* is the budget option for only 1 passenger for trips up to 10km. The cost is usually \$2 per kilometre, but is \$4 per kilometre during peak time (7am-9am and 5pm-7pm).
- *ZoomerStandard* is for 1-4 passengers and is for trips up to 20km. The cost is usually \$2 per kilometre, but is \$4 per kilometre during peak time (7am-9am and 5pm-7pm), per passenger.
- *ZoomerPro* is for 1-4 passengers with no maximum distance. It is the luxury option. The cost is usually \$3 per kilometre, but is \$5 per kilometre during peak time (7am-9am and 5pm-7pm), per passenger. There is also a surcharge of \$5 for each large bag.

You are to code the system for the call centre. When first run, it should present the user with a menu that has the following options:

1. Register a new customer.
2. Register a new Zoomer vehicle.
3. Book a new ride.
4. Display a report of all rides booked for the day.
5. Present a summary report for the month, that for each vehicle in the system shows how many rides were taken, total kilometres driven, and how much money was made.
6. Exit the program.

This menu should continue running in a loop until Option 6 is selected by the user.

The **Zoomer system** holds the following details about each **customer**:

- A unique **customer number**
- The customer's **name**
- A collection of all **rides** that the customer has taken

The system holds the following details about each Zoomer **vehicle**:

- A unique **vehicle registration number**
- Vehicle **make**
- The **Zoomer vehicle category** (e.g. one of Lite, Standard or Pro)
- **Total kilometres driven**

A **ride booking** consists of:

- A unique **ride ID**
- A reference to the **customer** who has made the booking
- A reference to the **vehicle** that has been booked
- The **date** and **time** of the ride
- The **start location**
- The **destination**
- The ride **distance** (in km)
- The number of **passengers**

For full marks, your design should contain all the following:

- **Inheritance** and the use of an **Abstract** class. You should also consider the use of an **interface** if appropriate.
- At least one **enum** type, used for an appropriate purpose (you can use more than one if you like, but you are only required to use one).
- **At least six classes** overall (including any abstract classes, and the Date class, if used).

**Code marking rubric:** Code design and correctness (50 marks total)

*1. Program functionality (25 marks)*

- Implementing overall main menu structure correctly
- Adding a new customer correctly to the system
- Adding a new vehicle to the system
- Adding a new customer ride booking
- Displaying the recent daily ride report (using appropriate calls to toString methods) as described
- Displaying the summary monthly report (using appropriate calls to toString methods) as described

*2. Design requirements (20 marks)*

- Sensible overall class design and adherence to good design principles
- Modularisation of complex tasks within classes
- Validation checks on user input
- Appropriate and correct use of inheritance (including abstract)
- Appropriate and correct use of enum type

*3. Adherence to coding standards (5 marks)*

- All code written should adhere to the guidelines set out in the FIT1051 Coding Standards.

**Suggestions for Implementation:** This is the biggest program you will have written for the unit! Make sure you approach it in a way that lets you tackle it bit by bit (i.e. problem decomposition). Suggested strategies include:

- Coding classes that do not rely on any other class first, and doing simple testing of them.
- Coding a simple menu and being able to make that work (and exit appropriately).
- Considering how menu items align with methods you write (and again how you break up the problem) and writing those, one menu item at a time, in order of complexity.

Do it bit by bit, breaking it up into logical chunks. And test... test... TEST!

**Submission Instructions:** Please submit your IntelliJ project folder as a .zip file and submit to the Assessment 4 link on the Moodle Assessments page as shown below. If you are not sure how to zip your project, please refer to the video here. **MAKE SURE YOU DOWNLOAD FROM MOODLE AFTER THAT TO CHECK IT IS THE RIGHT SUBMISSION!**

**Interview component:** You will be asked to demonstrate your program at an interview in Week 14/15, following the code submission date. You will be contacted by EMAIL to book an online interview time.

The interview will be up to 15 minutes in length, and consist of 6 questions. You may be asked to explain your code, your program designs, to modify your code, to discuss your coding decisions, or to explain the any of the coding concepts taught in Weeks 1-12 that this Assessment covers. Not being able to explain key design decisions or important Java syntax may result in failing the assessment.

Interviews will take place online via Zoom. You must have access to a stable internet connection and a working webcam, and your webcam must be switched on for the duration of the interview.

It is your responsibility to make yourself available for an interview time during Week 15, and to attend your online interview on time. The interview is worth 50% of your assessment mark, and any student who does not attend an interview will receive a fail grade for the assignment. Your interview must take place before the end of Week 15.

If you cannot attend your interview or miss your scheduled time, you **MUST** apply for Special Consideration.