Assessment Tool

Qualification national code and title	ICT40515 Certificate IV in Programming
Unit/s national code/s and title/s	ICTPRG401 Maintain open source code programs ICTPRG404 Test applications ICTSAS414 Evaluate System Status

Assessment type (☑):

☐ Questioning (Oral/Written)
☐ Practical Demonstration
☐ 3rd Party Report
☒ Other – program

Assessment Resources:

PC with installed IDE for programming (such as Netbeans for Java or Visual Studio for C#);
Internet access and an account on a Git host (such as GitHub or Bitbucket);
Access to Blackboard cluster Test Driven Development.

Assessment Instructions:

This assessment requires you to maintain and test an open source application. You are being assessed on elements 2 to 6 of ICTPRG401 and all elements of ICTPRG404.

Due date: End of week 16 for part 1 (program) and end of week 18 for part 2 (tests).

1. Complete all the assessment tasks below.
2. Observation by your lecturer of you doing the assessment is considered part of the assessment process.
3. Submit your documentation into the Blackboard assessments area.
4. All skills must be demonstrated to achieve a satisfactory result.
5. All work submitted must be your own individual effort.

Qualification national code and title	ICT40515 Certificate IV in Programming
Unit/s national code/s and title/s	ICTPRG401 Maintain open source code programs ICTPRG404 Test applications ICTSAS414 Evaluate System Status

Assessment Instrument:

Assessment 2: Develop and test open source application

For this assignment, you are required to review, modify and test an open source vehicle rental system application. You must use the Git source control tool in conjunction with an online repository such as GitHub.
There are 2 deliverables:

- Part 1: Modified open source application
- Part 2: Testing of the open source application.

Please complete the following tasks:

PART 1: Develop open source application

1. Set up and familiarise yourself with the project
    a. Review the starter code that is available for this open source system. (The code is in Java but you may convert it to C# if you prefer.) The starter code consists of classes for Vehicle, Journey, FuelPurchase and Service. Your lecturer will provide the GitHub repository location containing the starter code.
    b. Review the documentation and source code comments on the GitHub repository to better understand the project.
    c. Import the starter code into your own online repository. You will need to have a registered account with GitHub (or similar Git hosting service).
    d. Download the project to your local computer. Ensure that changes to your project directory will be tracked by Git source control.
    e. Build and run the application from within the IDE of the programming language.
2. Develop the application code
    a. Review the Issues that have been registered on the original GitHub project. Your task is to develop code to address the feature requests and issues/tasks.
    b. Under the test-driven development model, you should plan your tests upfront. (You can now write the initial unit tests as described in Part 2 task 1&2.)
    c. Update your local copy of the application source code to address the list of issues.
    d. Test your code changes, and commit them when each is ready. (You can now write the unit tests described in Part 2 task 3.) Follow the community participating standards defined in the original repository when writing the commit messages.
    e. Upload your code to your remote repository so that it is in sync with the changes committed to your local Git repository.
3. Project documentation and communication
    a. Update the documentation for this project by adding source code comments, updating the documentation files and creating a Wiki page. Do this work on the remote GitHub repository.
    b. Pull the changes made in the remote repository down to your local repository to bring it up-to-date.
    c. Notify the original/upstream repository owner of your project changes. (ie. Submit a Pull Request.)

Qualification national code and title	ICT40515 Certificate IV in Programming
Unit/s national code/s and title/s	ICTPRG401 Maintain open source code programs ICTPRG404 Test applications ICTSAS414 Evaluate System Status

d. Finally, participate in communications with the online community. For example, you can respond to a review/conversation, update an existing issue or create a new issue in the original repository identifying a possible future improvement to the system. Ensure that all communication exchanges with the online community are socially acceptable.

Submit a link to your online project repository and supporting screenshot evidence (of working program and community communication).

PART 2: Test the application

1. In your role as white box tester, you should now set up the unit testing framework/tool that you will use to automate the testing.
2. Plan your testing and prepare unit tests for testing that the original starter code works correctly. Execute those tests against sample test data.
3. Plan and prepare unit tests for testing that your functional enhancements work correctly. Execute those tests against sample test data.
4. Analyse the test results. Fix any code defects that were identified during testing and then re-test.

Submit your unit test code (or link to the online repository containing it) plus supporting screenshot evidence (test execution results).

Qualification national code and title	ICT40515 Certificate IV in Programming
Unit/s national code/s and title/s	ICTPRG401 Maintain open source code programs ICTPRG404 Test applications ICTSAS414 Evaluate System Status

Appendix: Development brief for Rental Car Management system

The development tasks are listed in the Issues tab of the GitHub project. There may also be TODO comments in the source code. Here are the main programming tasks that you must add code for:

1. **REQ 1: Main/UI Class(es):**
   - **a.** The user interface can be either a GUI (graphical user interface) frontend or console-based.
   - **b.** The main program must use all of the classes and methods to provide a fully functional, user-friendly application for viewing and recording vehicle rental data.
   - **c.** You may add or change methods as required to get your application to work. For example, the program should generate an error message when an attempt is made to rent a vehicle that is due for service.

2. **REQ 2: Rental Class:**

   You must create a new class (or classes) for calculating the rental cost. There are 2 ways of calculating the rental charge – either Rental Per Kilometers or Rental Per Day option.
   - For Rentals per kilometres, assume a default cost of $1 per KM travelled.
   - For Rentals per day, assume a default cost of $100 per day.

3. **REQ 3: Vehicle Class:**

   Amend the PrintDetails() method so that it displays additional information, including the rental revenue and number of kilometres since the vehicles' last service. Sample output:

   Vehicle: Holden Commodore 2014
   Registration No: 1 ABD 760
   Total Kilometres Travelled: 294.0
   Total services: 2
   Revenue recorded: $300
   Kilometres since the last service: 294
   Fuel economy: 10L/100km
   Requires a service: No