COMP2050 Coursework #1

Boon-Giin, Lee; Heng, Yu
boon-giin.lee@nottingham.edu.cn; zasshgyu@nottingham.edu.cn

University of Nottingham Ningbo China, School of Computer Science

1 Synopsis

Part of understanding how to improve as a software developer is learning how to reflect on new concepts and on your coding performance. In this coursework, you will be asked to write several aspects of the practical sessions of this course. The aim of this coursework is to ensure you have understood the principles involved in creating and developing maintainable software. To get started, click on this blog article on writing good, clean, clear and understandable code. Your answers to the tasks should be written in the style of a report. Please feel free to enhance your answers with snippets of commented code. The answers should be accompanied with the diagrams, figures or tables whenever possible and applicable.

IMPORTANT Make sure you understand what you are writing in your report. We reserve the right to briefly interview you if we think that you do not understand what you write about. So, when you write your report, do not simply copy/paste large portions of text from existing articles or other resources - as this does not demonstrate your understanding of the topic. You might also run into issues with plagiarism.

Table 1: Coursework Summary

Weight	25%
Format	Report with Maximum of 1000 Words
Submission Date	11th November 2019, Monday at 4 p.m.
Late Policy	Standard Policy
Submission Method	Electronic Submission via Moodle
Feedback Date	2nd December 2019 (Expected)
Feedback Method	Individual Comments via Grading System on Moodle

2 Deliverable

Your submission will be a PDF file submitted through Moodle on or before the given due date. This single file must contain all the answers. We will not accept submissions containing multiple files. Check nearer the submission date for the live link to the Moodle submission system. Furthermore, we might refuse to continue reading answers to individual questions beyond 1000 words (although we will not stop in the middle of a sentence, but try to stick to the limit). Please add the overall number of words you wrote for your answer to the front page. To mark your work, we may wish to print off a copy, therefore please add your full name and student ID to each numbered page of your completed report. Kindly use the word coursework 1 and your student ID as the filename for your submission, e.g., cw1-20031168.pdf. Failure to adhere to these rules may result in our refusal to mark your coursework.

3 Plagiarism

You are gently reminded that we are at liberty to use plagiarism detection software on your submission. Plagiarism will not be tolerated, and academic offenses will be dealt with in accordance with UNNC policy and as detailed in the student handbook. This means you may informally discuss the coursework with other students but your submission must be your own. Please also note that it is not permitted for you to copy and paste text from another source without correct referencing.

4 Assessment

This coursework is worth 25% of this module and as such is marked out of 25. Each of the 25 marks (each task) refers to a degree classification as outlined in Table 2.

Table 2: Marking Classification for Tasks and Presentation

Mark	Classification	Marking Schema
0	Fail	• No task was submitted, is missing or is nonsensical.
5	Poor	• Poor description of the answers, no reflective content, incorrect code snippets or did not answer the question. • Very poor presentation.
10	Average	• Basic completion of the answers with partially correct evidence but no linkage with lectures or laboratory exercises. • Adequate presentation but with inconsistencies and some errors.
15	Good	• Good completion of the answers with correct evidence but some problems either with commenting, minor inconsistencies, an attempt at reflective content, basic linkage with lectures or laboratory exercises. • Good presentation, mostly correct and consistent with minor errors.
20	Excellent	• Very good completion of the solutions, complete with a strong link to the lectures or laboratory exercises. • Explanations provided, correct and clear statement of reflection and implications of the solutions. • Great standard of presentation, consistent and clear use of code, style, diagrams (and/or tables and/or figures) and writing.
25	Outstanding	• Excellent completion of the solutions, with correct linkage to the lectures or laboratory exercises. • Solutions provided which go beyond those described in the lectures or laboratory exercises, with fully referenced sources. • Flawless standard of presentation and could be externally published.

5 Task

You are given the source code of an existing project - a device manager server for connected cars. This code has been modified to provide lots of opportunities for maintenance and refactoring. Your task is to understand and identify the problem in the code, then apply what you have learned from the lectures, laboratory exercises, workshops or other reading materials to perform maintenance on the provided source code. You must write up your solutions to **FOUR** tasks listed below. Each solution will have an associated word count which you must also not exceed, which is 250 words excluding illustrations (diagram, figures, tables etc.) and supporting code snippets.

5.1 Task 1

Write any missing code in main method such that the program will execute. You should pay special attention on how to read the input from the excel files. Provide several outputs to demonstrate how the program works. [15 Marks]

The attached folder contains four Excel files (ChangeState, Create, Delete, Mode). [10 Marks]

(a) What do these files represent?
(b) Explain the meaning of the contents of the file Create.
(c) Show how we can read the values of these files.

5.2 Task 2

Explain the overall purpose of this project. What is it about? What does it do? [15 Marks]

The class Device extends Observable, Runnable. [10 Marks]

(a) Explain what is Observable, Runnable.
(b) Why they are used in this project.

5.3 Task 3

Use Papyrus to draw the class diagram for the attached classes showing the followings: [25 Marks]

(a) Methods of each class.
(b) Attributes of each class.
(c) Inheritance.
(d) Aggregation.
(e) Access specifiers for each attribute and method.

5.4 Task 4

In the project, polymorphism is used in more than once. [10 Marks]

(a) Identify all polymorphism cases by identifying their classes and methods.
(b) Take one case and explain the need for polymorphism.

The project uses Collection classes as containers for certain values. [10 Marks]

(a) Identify the classes in which collections are used.
(b) What are these Collections.
(c) Describe what they contain.

Explain why some attributes in class Car are defined as final static. [5 Marks]

5.5 Tips & Hints

- It might be a good idea to start with reverse engineering a high level class diagram from the source code provided. This practice can provide you a clear understanding and allow you to explain the source code appropriately (e.g., how you identify the relationship and the type of relationship between classes), can give a positive impact on your mark. Providing only class diagram without explanations are not sufficient to give you mark.

- You should give explanation of why your code alteration can provide improvement to the project for every task you completed. Providing solutions to the tasks without explanations are not sufficient to give you mark. Also, providing similar type of tasks will not bring you any benefits. You must tackle for different type of the problems in the source code.

6 References

If you use any text or information from any resources such as books, journals, articles, proceedings etc., you have to provide those references. A guide for references and citation can be found here.