

# COMP5911M Advanced Software Engineering

## Coursework 2

This assessment is based on material from the lectures dealing with components and software metrics. There is only one task with four subtasks, with a total of 100 marks available. You should submit all your answer using Gradescope (link in the “Submit my work” folder in Minerva).

This assessment is worth 15% of your overall grade for this module.

### Task 1

Download task1.zip from the ‘Coursework 2’ folder in Minerva. Unzip task1.zip and then delete it. This should leave you with two directories named before and after. These directories contain code for the Car Rental example used in the lectures and exercise, before and after the refactorings of Exercise 4 have been applied.

Task 1.1 (20 marks) Compute the following metrics for source code provided in the before directory:

- Total SLOC (Source Lines of Code) in the package
- Average number of methods in a class
- Maximum method complexity for each of Car, Rental and Customer
- Instability for each of Car, Rental and Customer
- Abstractness of the package

Task 1.2 (20 marks) Compute the same set of metrics for the code in the after directory.

Task 1.3 (30 marks) Explain how you computed each of the metrics in task 1.1

Task 1.4 (30 marks) Compare the values of the metrics before and after refactoring. Relate this to what has been achieved by the refactoring. Do the metrics tell us anything useful about how the software has changed? Are they misleading in any way?

PLEASE NOTE:

- SLOC is defined as the number of non-blank, non-comment lines. You can compute this manually or using a tool such as David Wheeler’s sloccount (<https://dwheeler.com/sloccount/>)
- Use McConnell’s simplified approach to compute complexity.
- Do not include unit testing code in your calculations.
- Do not use more than two decimal places when quoting non-integer results.