

# COMP201 Assignment 2

Assessor: Dr. T. Carroll

## 1 Key Details

**Module** COMP201 - Software Engineering I

**Weighting** 20% of module grade

**Deadline** 12 noon, Thursday 12th December 2019

**Purpose of Assignment** To assess the student's ability in understanding requirements, in following object-oriented design principles, and in implementing an object-oriented design in Java.

**Learning Outcomes Assessed:** The following learning outcomes are assessed, as per the module specification:

LO4 Be fully aware of the principles and practice of an O-O approach to the design and development of computer systems;

LO5 Be able to apply these principles in practice;

LO7 Be able to demonstrate how to effectively implement an O-O design in an O-O language such as Java or Python;

## 2 Informal System Requirements for a Garage

You have been approached by a garage to design a computer system to log jobs.

- The garage will perform inspection tasks, repair tasks, and maintenance tasks on different types of vehicle: cars, vans, and busses.
- The garage carry out the following inspection tasks: the MOT test and the general diagnostic test (for when a customer does not know the problem with their vehicle).
- The garage carries out the following repair tasks: body repair, engine repair, window replacement, and insurance mandated repair.
- The garage carries out the following maintenance tasks: air conditioning top-up, body respray, and tyre change.
- The shop has multiple staff members working each day: the receptionist, the manager, and several mechanics. The manager is in charge of all staff working that day, though they are also a qualified mechanic, and thus works on similar tasks to the mechanics when short-staffed.
- The workflow of the garage is as follows:
  - When a customer comes into the shop, they park their vehicle, and walk into the reception office, where they discuss their need with the receptionist (who has domain specific knowledge, and can therefore give the customer the correct advice).
  - The receptionist then opens a ticket, which links the customer, their vehicle, the work required, the deadline, and the quoted price. The ticket's status is currently *waiting*. The customer then leaves their vehicle at the garage.

- At the start of the day, each mechanic uses the system to view the *waiting* tickets, and gets the first one available. They update the ticket status to *in progress*.
- The mechanic collects the vehicle, and performs the work required. If an unexpected cost is encountered, then this will be placed on the ticket notes.
- After the work is carried out, the ticket status on the system is changed to *check* by the mechanic. The mechanic then proceeds to the next ticket.
- Throughout the day, the manager uses the system to view the tickets which have the status of *check*. They will look at each ticket, and then check that the work has been carried out to a good standard. If needed, the manager will update the price in cases where unexpected costs were encountered. Once the manager is satisfied, the status of the ticket is moved to *signed off*.
- Throughout the day, the receptionist will use the system to look at jobs that have the status *signed off*, and telephone the customers to inform them that their vehicle is ready to collect.
- The customer comes to the garage, and is presented with the bill. Once they have settled the bill, the receptionist will then show the customer to their vehicle and the customer drives off.
- Once the customer has left, the receptionist changes the status of the ticket to *complete*.

### 3 Assignment Tasks

The 5 tasks detailed below are based upon the informal system requirements that are outlined in Section 2.

**Task 1 (20%)** You are to create 4 separate lists, each with added details if required.

**1a)** List all candidate classes, their candidate attributes, and their candidate operations.

**1b)** List all potential inheritance relationships.

**1c)** List those candidate classes that are to be eliminated, and give justification as to why.

**1d)** Give the final list of candidate classes, along with their attributes and their candidate operations. You should ensure minimal data duplication (e.g. if a customer has multiple cars in for repair).

**Task 2 (20%)** Produce CRC Cards for each class. For each CRC Card, comment on whether the class is “Good” or “Bad”, and give justification for your reasoning. If it is “Bad” then you should state how it may be improved, but do not implement this improvement.

**Task 3 (20%)** Produce a UML Class Diagram showing the classes, attributes, operations, and associations of the system (use answers from Task 1 to guide you). You should be sure to use the correct type of association, navigability, and multiplicity.

**Task 4 (20%)** Produce **skeleton classes**<sup>1</sup> in Java for each of the classes identified by the UML Class Diagram. Each class should have the correct inheritance, attributes, and operations. You should create **only empty** methods in the Java class, using a dummy return statement, if required. The classes should all compile, but there is no need for a main method.

**Task 5 (20%)** Produce a UML activity diagram capturing the workflow of the garage.

---

<sup>1</sup>Classes that set out the attributes and methods of the design, that do not actually contain any implementation, and that use dummy code that allows the code to be compiled.

## 4 Deliverables

You are required to deliver the following items:

1. PDF Format Document containing the diagrams and written answers for Tasks 1,2,3, and 5. It should have your Name, University Username, and Student ID at the top. For the diagrams, you may use software, and you may hand draw them and scan in the image, but it MUST be legible, otherwise you may be awarded 0 marks.
2. Java Source code files (\*.java files) for the answer to Task 4. Source files should contain your Name, University Username, and Student ID at the top as comments.

## 5 Submission

You are to submit a zip file ("**studentID.zip**", where **studentID** is your numerical 9-digit student ID number, found on your Student ID card) electronically via SAM, no later than the deadline stated in Section 1. This zip file should contain that which is detailed in Section 4.

You can access the SAM submission system at the following URL: <https://sam.csc.liv.ac.uk/>.