Assessment Brief Proforma

1. Module number	*SET10112/SET10412*
2. Module title	*Formal Approaches to Software Engineering*
3. Module leader	*Peter Chapman*
4. Tutor with responsibility for this Assessment Student's first point of contact	*As above*
5. Assessment	*Practical coursework*
6. Weighting	*75% of module assessment*
7. Size and/or time limits for assessment	*10 pages in LNCS format (template supplied on Moodle)*
8. Deadline of submission	Your attention is drawn to the penalties for late submissions ***Report:*** *29th April 2022 before 1500* ***Demo:*** *6th May 2022 before 1500*
9. Arrangements for submission	***Report:*** *Upload your PDF (more details given later) to the Moodle submission page.* ***Demo:*** *Upload your video file (more details given later) to the Moodle submission page.*
10. Assessment Regulations	All assessments are subject to the University Regulations

11. The requirements for the assessment	*See following document*
12. Special instructions	*None*
13. Return of work and feedback	*Marks and general feedback provided by* ***20th May 2022.***
14. Assessment criteria	*See following document.*

Coursework - Self-driving cars!

Peter Chapman

SET10112/SET10412

Abstract. In this coursework, you will have to design a "driver's assistance" system for an automated car. You will then create Ada-SPARK specifications and bodies for your system. Extensions to the given specification are encouraged, and will enable you to gain higher marks. A report, containing your specification definitions and descriptions, must be produced that is no longer than 10-pages in the LNCS style template. These templates are made availabe for you on Moodle.

1 Overview

Statistically, cars are dangerous. Human error contributes to most accidents on the roads. You are tasked with making car-travel more safe. You do not require a multi-billion pound company, or even access to a car itself: you need only use what you have learnt in this module.

You will develop a driver's assistance system for an automatic-gearbox electric car. This system will prevent certain actions when they are unsafe. For example, the driver will be unable to start the car unless it is in the Parked gear; the speed limit must be obeyed at all times; the gearbox cannot go directly from Drive to Reverse; and so on. In particular:

- The car cannot be turned on unless it is in Parked.
- The car cannot be driven unless there is a minimum charge in the battery.
- Once in motion, the system will warn of low charge.
- The speed limit can never be exceeded.
- The speed of the car must be zero in order to change gear.
- If the car's sensors detect an object, then the car cannot move towards that object.
- The car must have a diagnostic mode which renders it incapable of any other operation.

Note that these are the *minimum* requirements for your system. You are encouraged to add your own requirements as extensions.

1.1 A Note on Extensions

Extensions are of two types. The first type include those which introduce new complexity to the formal part of the system. The second type include those which will aid you with demonstrating the correctness of the formal part of the system. Both are valid, and full marks cannot be achieved without having both types present.

1.2 SPARK levels

Recall from the lectures and materials online that there are grades of SPARK implementation:

- **Stone level** - valid SPARK
- **Bronze level** - correct initialisation, data and information flow
- **Silver level** - absence of run-time errors
- **Gold level** - proof of key integrity properties
- **Platinum level** - full functional proof of requirements

Each level subsumes the level below it, in that, for example, one cannot achieve silver without achieving bronze. Normally, these levels are applicable at the program level, but you will be assessed at the subprogram level. For example, your "gear change" system may achieve bronze, but your speed system may achieve gold.

2 Report

Your report *must* use Springer's Lecture Notes in Computer Science (LNCS) template. Templates for LaTeX and Word are provided on the Moodle page for the module. The page limit is **10 pages**. Deviation from either the template or the page limit will result in a penalty on your coursework. (For reference, this document is in LNCS format.) Your report should include the following sections:

1. *Introduction* - a high-level overview of the problem you have solved, including brief descriptions of your solution and any extensions to the problem. This section, if read alone, should give the reader a clear, if brief, picture of what you have done.
2. *Controller Structure* - a high-level view of your driver-assistance system. You should include a description of why your system is structured in that way (i.e. explain any global variables, types, and private clauses in your system.)
3. *Descriptions of procedures and functions* - a more detailed look at the individual components, including descriptions of parameters, constants, post-conditions, pre-conditions, and initialisations.
4. *Proof of Consistency* - you should include descriptions of which parts of your system are formally verified, justifiying your decisions. For anything that is particularly key to your specification (and in any case for at least one procedure or function) you should demonstrate how the proof obligations are satisfied. In other words, prove the obligation by hand.
5. *Extensions* - you should describe the extensions you have added to the basic specification.
6. *Conclusion* - this section should detail any shortcomings of your specification, including aspects you would like to have included but did not.

3 Submission

The assessment will be consist of two parts:

- the **pdf** version of your report. **Do not submit Word or TEX files**. The name of the pdf should be your student number. You **will not** submit your code to Moodle.
- the **video demo** of your working code, submitted one week after the report. The demo will allow the marker to determine which aspects have been implemented, and which SPARK level you have achieved on each aspect. Whilst the demo carries no marks, failure to complete it satisfactorily will result in a coursework mark of 0. The type of video file is limited to those which are natively supported by browsers. (i.e. .mov .mp4 .m4v .ogv .webm)

4 Mark Scheme

The coursework will be marked out of 100.

> **Any deviation from the template or page limit will limit your mark to** 65

The breakdown of marks will be as follows:

- *Solution Structure* **15 marks** - does your program demonstrate good software engineering design principles, such as re-usability?
- *Solution Coverage* **25 marks** - does your program fully address the aircraft scenario? Are your subprograms appropriate, with appropriate variables, visibility etc.? Do the procedures and functions make sense in the context?
- *Solution Correctness* **25 marks** - which SPARK level have you achieved? Are any pre-conditions too strong, and if so, why? It is this section which the **demo** will primarily assess. The SPARK levels do not map on to a linear scale of marks.
- *Solution Extensions* **25 marks** - in what ways have you extended the given problem? This part of the mark scheme is intentionally under-specified.
- *Quality of written report* **10 marks**- adherence to the template, quality of written English, quality of any displayed formulae or diagrams, document flow (i.e. does your work read like several sections which have little in common, or like a coherent narrative?)