**2805ICT System and Software Design**

**3815ICT Software Engineering**

**7805ICT Principles of Software Engineering**

This is an assessed workshop

**WORKSHOP 3**  **UML & first stage assessment**

Student name: Nguyen Quang Huy Student ID: s5257464 Enrolled Course Code: 7805ICT

**1. Cohesion & Coupling (15pts)**

1. What is cohesion, and what is coupling (5pts)?

* Cohesion: The degree to which elements of a component or object interdepend.
* Coupling: The degree to which components or objects depend on one another

1. To improve the quality of a software system, we need to increase or decrease cohesion? Use examples to explain bad and good design regarding to cohesion (5pts).

* To improve the quality of a software system, we need to increase cohesion.
* Example of good design: frequently used methods (funcs) are defined inside classes (in the same file), which improve code reusability and readability
* Example of bad design: methods/funcs are not defined in whatever file(s) it is needed, this decrease code reusability since the method must be defined multiple times.

1. To improve the quality of a software system, we need to increase or decrease coupling? Use examples to explain bad and good design regarding to coupling (5pts).

* To increase the quality of a software system, we need to decrease coupling
* Example of bad design: a system with multiple (a lot) of dependencies like multiple sub-classes, changing something in Module A might cause a ripple effect that affect other modules (childs) in an undesired way(s)
* Example of good design: Modules are clearly separated, there are less dependencies

**2. Requirement Engineering (15pts)**

1. What are the four common software requirement defects? (5pts)

* Incomplete, ambiguous, too weak, too strong

1. What is a formal way (formular) to write high quality requirements? Rewrite the requirement below by using this way (10pts)

The formal way to write: User preconditions, events, post conditions and constraints

“*A user can operate the microwave oven by pressing the start key to make it start to cook for one minute*”.

As an user, I can press the start key to start the “one minute cooking procedure” of the microwave oven

**3. OO design principles explanation (10pts)**

1. Explain the meaning of encapsulation, inheritance, and polymorphism with examples in OO design (5pts).

* Encapsulation: The combining of attributes and behaviour into a single object whereby the actual implementation details are hidden.
* Inheritance: The ability of an object to indicate that it will acquire the attributes and behaviour of some other objects.
* Polymorphism: The ability of different objects to perform the appropriate behaviour in response to the same message

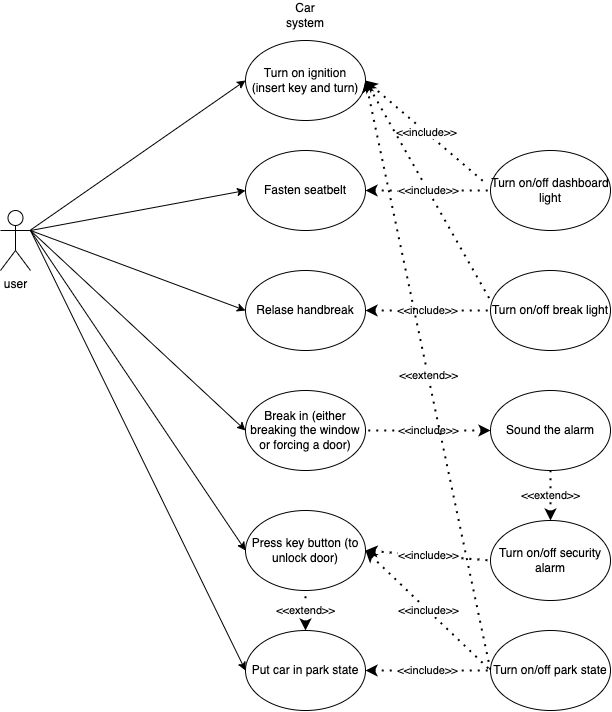
1. Discuss the benefits and disadvantages of them (5pt).

**4. Use case diagram (15pts)**

Based on the requirements given below to draw a use case diagram. Marking criteria:

1. All the use case diagram elements are included correctly.
2. All functional requirements are covered
3. All use cases are named properly (as verb + noun )
4. Correctly use include and extend relationship





**5. Class diagram (20pts)**

Based on the requirements given in question 2, draw a class diagram. Marking criteria:

1. All functional requirements are covered, all classes are identified.
2. Suitable methods should be given.
3. Correctly use UML notations.
4. You need to create a class Light as parent class for BrakeLight and DashboardLight
5. Should include correct multiplicity
6. Should include all three different types of relationship: aggregation (composition), generalization, and association
7. Use CapitalCamelCase for class names and lowerCamelCase for method names.

Diagram

Description automatically generated

**6. Sequence diagram (15pts)**

Based on the requirements given in question 2, draw a sequence diagram to describe the scenario: the driver press key button to switch the alarm off and unlock the door, and then the driver open door to enter the car and put the car to park. Marking criteria:

1. Correctly use sequence diagram notation.
2. The class names and method names must match what in the class diagram.

**7. Interview (10pts)**

The workshop teacher (assessor) will arrange a one-to-one interview at week 10 or 11 workshop time. During the interview, the assessor may ask you to explain your answers and discuss the marking. The interview will also cover your work in the assignments. The purposes of the interview are:

1. Help you to clarify marking issues.
2. Check your understanding.
3. Check if the submitted work was done by you and other possible academic integrity issues.

If the assessor suspects that some of the submitted work was not completed by you, it might raise a concern and might deduct your marks accordingly.

**8. Additional exercises for 3815ICT and 7805ICT**

This question is not assessed, however, if you don’t complete this question, your other questions will not be marked (the question is only for students who enrol in 3815ICT and 7805ICT).

Write 15 lines of a reflective report on the previous activities. Analyse and evaluate their relevancy to your future work.

**9. Additional exercises for 7805ICT**

This question is not assessed, however, if you don’t complete this question, your other questions will not be marked (the question is only for students who enrol in 7805ICT).

Design an open-ended question (that means there may be several correct answers) that could be suitable for

1. A final exam
2. A job interview for software engineer.