ASSESSMENT TASK 1: DESCRIPTION

This task assesses your ability to plan, design and start developing a car traffic simulator - the details of which will be released during the first part of semester. You are expected to:

Document the software development process;

Present the code design in the form of UML class diagrams; and

Establish a GitHub repository and start implementing the code using the UML class diagrams.

The quality of your code is assessed in terms of general readability, application of Java coding standards, and your ability to apply good coding practices as discussed during the subject.

Using the software development process illustrated below, students plan, design and develop a Car Traffic Simulator in Java using IntelliJ IDEA connected to a GitHub repo (share this with your practical class supervisors!).

Overall outcomes:

Each student incrementally produces their own project plan, design and solution

The project plan consists of a working document based on the software development process – the document incrementally answers questions about the problem specification, decomposition, class design, data design, method design and algorithm design

The design is represented as UML class diagrams using Violet UML editor: http://alexdp.free.fr/violetumleditor

Each student uses a GitHub repo to incrementally make changes to the working document, UML design, and Java source code

The working document

To plan out the software development process, your working document is expected to describe the following:

Problem specification: What exactly is the problem? How will the program be used? How will the program behave?

Problem decomposition using UML class diagrams: What objects will be used and how will they interact?

Divide the problem into objects

The UML class diagrams should answer the following design questions:

Class design:

What role(s) do objects of this class perform?

What member fields do objects of this class need? Should they be public or private?

What methods do objects of this class need? Should they be public or private?

Method design:

What should its method signature be?

What task will it perform? What algorithm will it use?

You are not expected to produce a complete development report – that’s why it’s called a “working document”. However, you need to at least produce the following outcomes for this first part of the project work:

Produce a working document that illustrates your designs for approximately 4-6 classes – one of which will be the Main class that contains the main() method

The classes should be drawn in a UML class diagram – the diagram should include information about class names, member field access control, method access control and class relationships (you get marks for using the correct UML notation for these things)

Pick at least 3 classes and start implementing them – include appropriate test classes that check and verify that these 3 production classes are working as expected - there is no need to create a test class for the Main class!

You get marks for ensuring that your class code matches your UML class diagrams – there is no need to include the test classes in your UML class diagrams!

**For example, let’s say that you create some classes that represent objects such as a car, a traffic light and two road objects (this is a very simplified version of the entire problem domain). Each road might start out as one-way only. The outcome of your program could be to have the car object move from the first road object to the second road object. You could place the traffic light object at the end of the first road object.**

**You might treat the Main class as the simulator for the time being. The simulator needs to run a sequence of looped updates to change the state of the car and the traffic light. While the simulator is running, the car moves along the road and the traffic light changes colour. If the car reaches the traffic light at the end of the first road then it must stop. When the traffic light turns green, then the car moves onto the second road and continues to move until it reaches the end of that road.**

Submission Requirements

Please submit the following items to this assessment task:

A link to your project work GitHub repo

A zip of your project work repo

Please include your UML class diagrams and your working document inside your IntelliJ Java project - place them in a subdirectory within the project folder called "design".

Please note that you may include more than one UML class diagram. It is sometimes hard to say everything you need to say about the classes and class relationships with only a single UML class diagram.

Name:

Section 1

Problem specification: What exactly is the problem? How will the program be used? How will the program behave?

Section 2

Problem decomposition (using UML class diagrams): What objects will be used and how will they interact?

1. Divide the problem into objects
2. The UML class diagrams should answer the following design questions:
   1. Class design:
      * What role(s) do objects of this class perform?
      * What member fields do objects of this class need? Should they be public or private?
      * What methods do objects of this class need? Should they be public or private?
   2. Method design:
      * What should its method signature be?
      * What task will it perform? What algorithm will it use? **(can use pseudocode to illustrate point)**

(**a car, a traffic light and two road objects**) – what the video said

\*just do what you can; focus on what you can\*

Test would check setters and getters and methods for movement (car - vehicle)

Design ideas: (supplied by video)

Check onenote\*\*