SOFT252 Coursework FAQ

**Question:**

What does it mean in the scenario when it says *Cars booked from the pool are booked for whole days, so, for example, if a member of staff needed a car, he/she would access the system on the day it was required and select a car from those available. The car would then be allocated to that member of staff for that day only.*

**Answer:**

The above text has been replaced with: *Cars booked from the pool are booked for whole days, so, for example, if a member of staff needed a car, he/she would contact the Transport Office who would access the system on the day it was required and select a car from those available. The car would then be allocated to that member of staff for that day only.*

**Question:**

How are marks allocated to the various parts of the assignment?

**Answer:**

The following allocation has been added to the specification:

### *Allocation of marks within this assignment*

*The design elements are worth 50% of the assignment mark. These elements are:*

1. *UML Class Diagram.*
2. *Development of JUnit tests.*
3. *Reflection of your design.*

*The software implementation elements are worth 50% of the assignment mark. These elements are:*

1. *Java Class Library containing your data model implementation*
2. *Java GUI Project with appropriate HCI that uses the class library*
3. *The Javadoc web site produced from your class library*

**Question:**

Two projects to be submitted, what is in each one?

**Answer:**

Consider a situation where you are in a working environment. A particular scenario has been identified that requires a data model and may have a number of different application based upon it, for example the transport office staff accessing car details, another application (maybe on an intranet) allowing staff to view/book pool cars. So the same data model classes might need a number of different applications. Of course you wouldn’t copy and paste the classes from one project to another, would you. So put all the data model classes and any design pattern-based classes/interfaces etc. into one project. Your GUI classes would then be in a separate project that accesses the data model project. Review Tutorial 6 to see an example (this creates a utility library but the principle is the same). Your class library will also have in it the JUnit tests.

**Question:**

Javadoc

**Answer:**

Again, we are considering the working situation where you might be one of a tem. What would YOU want to know if you were using the data model? All data model classes need javadoc, the GUI classes do not. Getters and setters need to be included – although they will mostly be straightforward, a developer needs to know this without looking at the source code and that developer might not have been involved in creating the data model.

**Question:**

JUnit testing

**Answer:**

See the tutorial!!

**Question:**

Class diagram

**Answer:**

As Pushpa has suggested, you need to finalise the class diagram once the data model has been developed, without the interfaces and classes etc that are related to the design patterns you have used. However, your reflective report will discuss the design patterns you have used and you should include in it some mini class diagrams showing the design pattern implementations.

**Question:**

How does the project containing the data model classes link up to the GUI project

**Answer:**

See tutorial 6 section 6 to see how to access one project from another. In the case of the coursework, the GUI project needs to refer to the data model project.