# COMP258 Object Oriented Programming Assignment

## Value Due Date

10% 9am, Monday 9 June 2014

This is an individual assignment.

## Aim

Create a Processing application for a Road Code Quiz.

The Road Code Quiz should have 10 questions. The user is prompted firstly to start the quiz, and then to answer each question one at a time. After a response is given to each question, the user is told that they either (i) answered the question correctly or (ii) answered incorrectly, in which case they are shown the correct answer.

At the end of the quiz the user is shown a summary of how well they did in a graphical way using a gauge.

For an example of a quiz with the same specifications, take a look at this link:

<http://www.aa.co.nz/drivers/driving-school/road-code-quiz/>

## Steps

The following steps are a likely good order in which to do things. You can use/change as you need, as long as you fulfill the aim above.

**1. Make up your questions**  
You should make up your own questions based on the NZ road code, which can be found online here: <http://www.nzta.govt.nz/resources/roadcode/>

**2. Obtain some images** **to enhance your app**  
You are permitted to use images from the road code link above (use of which will *not* be considered plagiarism and they don’t need to be referenced).

You are also permitted to use any images for graphics and icons (e.g. ticks and crosses) obtained via Google image search with usage rights set to “Labeled for reuse”.

**3. Design a Button class**  
I have shown you an example of a Button class in lectures, but do not use that example. Instead, design your own Button. Note that my example lacks many sophisticated features such as responding automatically to mouse rollovers and having images as backgrounds. See the “Start quiz” button on the example quiz above.

**4. Design a Gauge class**

A Guage object is used to display the user’s final score out of ten at the end of the quiz. A minimal Guage would be a circle, a needle, and some labels positioned around the guage so that the final score can be easily read.

**6. Design a Question class**

A multi-choice question consists of (i) text, (ii) an image, (iii) four possible answers, (iv) a specification of which answer is correct, and (v) a state (either answer shown or answer hidden).

**7. Design a Quiz class**

A Quiz is a collection of Questions. It should have properties for both the user’s current score (number of correct responses) as properties describing which part of the quiz (e.g. introduction page or question number) is currently being displayed.

**8. Put it all together**

Once the classes are designed and tested, you can put them all together into a finished app. Make sure you test the app thoroughly before submission!

## Use of Inheritance/Polymorphism

As it stands, the above program can be constructed without inheritance and polymorphism. However, for full marks, use inheritance to make your quiz ask two different types of question: (i) multiple choice questions where the user chooses A, B, C or D, and (ii) short answer questions where each question is a single word answer typed in by the user. The short answer question will require a new control, namely a text box. Your program should have an abstract superclass for both classes of question.

## Marking Schedule

Marks for this assignment will be assigned in the following way:

* Steps 1-8 – 50%
* Extension using inheritance to multiple question types – 20%
* Good design of classes, properties and methods; and quality of code (naming, comments etc) – 15%
* Design, appearance and professionalism of the UI – 15%

Note that you **must use object oriented program (classes, objects, methods, inheritance etc) to solve this problem**. You could write a quiz program without proper classes (i.e. COMP103 style) but you will be penalized significantly if you do that.

## Submission

Submit a zipped version of your Processing project to the moodle handin box by the due date.