**Summative assessment: System implementation**

The goal of this project was to create code designed to support a driverless vehicle.

The functions I chose to support were:

Speed sign detection – The car will detect a speed sign, read the limit and adjust its target speed limit. If there are no further obstacles, the car will travel at this speed. This speed control module allows it to accelerate or decelerate to achieve the desired speed.

Traffic light detection – To detect a traffic light, distinguish its color and react accordingly. If red, the car would temporarily stop, wait for a green light, then accelerate back to its previous speed. If green, the car would continue to travel.

Object detection – The car can detect and gather information about objects in the road ahead. If deemed a collision risk, the car would perform an emergency stop. If not a risk, the car would continue to travel. Regardless of risk level, the object information would be recorded and can be later sent to the manufacturer for future development.

In order to code the above functions, I chose to split them into 3 separate modules; the Speed control module, the sensor module and the primary module.

The sensor module contains a Sensor superclass containing the car details. The individual sensor subclasses all inherit from this superclass. Each subclass allows us to instantiate objects to represent either a traffic light, speed sign or a physical object.

The speed control module contains the speed control class which allows us to instantiate a speed control module object. This object contains all speed related methods.

The primary ‘car’ section contains a car class which can instantiate both the obstacles listed above and the speed control module. It allows us to call on the required methods to adjust to the obstacles detected. It is from this file that our simulation is ran.

**The simulation**

The simulation itself is ran from the file called ‘MainFile’. First, you are prompted to hit enter to start up the car and begin the journey. During this simulation, the user is required to input the information that, in real life, would be fed to the program via the sensors. However, since that is outwith the limits of this code, the data must be manually fed.

First, you must input the initial target speed. In a real situation this would come from the GPS, and journey data. After this, the car will accelerate up to the target speed and continue travelling until it receives further inputs.

The inputs are as follows:

1 to simulate a speed limit sign, after which the user is prompted to input the limit on the sign. The car will then adjust its speed to the new limit.

2 to simulate a traffic light. The user is prompted for the color (Red or Green). If Green, the car continues. If red, the car will stop and await user input signalling a green light. When green, the car resumes travel at the prior speed.

3 simulates an object in the road. The user is again prompted for a description and the risk.

4 will end the journey.

Any other input is not recognised and therefore causes an error and prompts the driver to take over.

**Testing**

In order to test the software, I tested it both in its individual modules and as a whole. I used the python assert statement at regular intervals to confirm the integrity of my code. Please see code for some examples of these assertions.