PPA Assignment 3

This coursework is designed to test the content from Topics 1, 2, 3, 4 and 5.

Do you think there is a problem with any of the content below? Let us know immediately at programming@kcl.ac.uk.

For this week's assessment, consider the following scenario, and then complete the tasks that follow it:

We would like to create a simple simulation of a car race around a race track. There is an average time, in seconds, that all cars take to complete a lap around the track. The total time each individual car takes to complete a single lap is affected by the attributes of the car and the conditions of the race track. During a race, we would like to know who the leader is at the end of each lap. At the end, we would like to know the winner.

1. Model this scenario based on the following requirements (note that you should use the `int` data type for variables which store an amount of seconds):

    1. Create a class to represent a `RaceTrack`. A race track has an `averageLapTime`, given in seconds, which is the average time it takes for any car to complete one single lap around the track. A race track also has a flag `isRaining`, indicating if it is currently raining. When we create an object of this class, we should be able to supply values to these attributes. (1 mark)

    2. Create a class to represent a `Car`. When we create an object of this class, we should be able to supply values for the following attributes:

        1. an `id` number to identify the car,

        2. a `fuel` level, given in integer units with a maximum of 100,

        3. a `lowFuelBoost`, which is the amount of seconds in one lap that a car goes faster

when it's fuel level is low,

4. a `highFuelSlowdown`, which is the amount of seconds in one lap that a car goes slower when it's fuel level is high,

5. a `fuelConsumptionPerLap`, given in integer units, to indicate how much fuel a car uses up per lap,

6. a `pitStopTime`, which is the amount of seconds it takes for the car to complete a pit stop (to refuel the car),

7. a `rainSlowdown`, which is the amount of seconds that a car slows down in one lap when it is raining,

8. a `totalTime`, which is the total amount of seconds the car has taken during an *entire* race, and which starts at zero.

(1 mark)

3. In the `RaceTrack` class, create a method `determineRaceLeader`, which determines and returns the car (out of the three cars) that is the leader of the race. (2 marks)

4. In the class `Car`, create a method `completeLap`, which returns the total time, in seconds, that the car takes to complete one lap around a provided race track. This total lap time must be calculated using the following guidelines:

1. The base time for one lap is the race track's average lap time. (1 mark)

2. When the fuel level is above 50 units, the car is heavier and therefore, goes slower in this lap by the amount of seconds specified by `highFuelSlowdown`. Otherwise, the car goes faster in this lap by the amount of seconds specified by `lowFuelBoost` (1 mark).

3. When it is raining, the car goes slower in this lap by the amount of seconds specified by `rainSlowdown`. (1 mark)

4. At the end of a lap, the car uses up the amount of fuel given by `fuelConsumptionPerLap`. (1 mark)

5. The car will need to take a pit stop when the fuel level drops below the amount of fuel that the car requires to complete one lap. When the car takes a pit stop, this makes the car go slower in this lap by the amount of seconds specified in `pitStopTime`. This also refuels the car to full capacity. (2 marks)

2. Create a class `RaceSimulator`, which can be compiled and run from the

command line. Use this class to do the following (in order), using the classes and methods you have created for Question 1.

1. Create a race track and name the variable holding it *silverstone*. Set the average lap time of this track to *90* seconds and make sure it is not raining. (1 mark)

2. Create three cars, which have the following attributes:

   - Car 1:
     - the id is *1*,
     - the starting fuel is *55*,
     - the low fuel boost *6*,
     - the high fuel slowdown is *5*,
     - the amount of fuel consumed per lap is *25*,
     - the time taken for a pit stop is *12*,
     - the additional time taken for a lap when it is raining is *15*,
     - the total time starts at *0*,
   - Car 2:
     - the id is *2*,
     - the starting fuel is *60*,
     - the low fuel boost *8*,
     - the high fuel slowdown is *7*,
     - the amount of fuel consumed per lap is *28*,
     - the time taken for a pit stop is *14*,
     - the additional time taken for a lap when it is raining is *10*,
     - the total time starts at *0*,
   - Car 3:
     - the id is *3*,
     - the starting fuel is *90*,
     - the low fuel boost *10*,
     - the high fuel slowdown is *6*,

- the amount of fuel consumed per lap is *30*,
- the time taken for a pit stop is *16*,
- the additional time taken for a lap when it is raining is *9*,
- the total time starts at *0*,

(1 mark)

3. Make your cars race for two laps around silverstone, and after each lap, print the id of the leader of the race. (1 mark)

4. Make it rain on silverstone. Then, make your cars race for one more lap and, finally, print out the id of the winner of the race. (1 mark)

Once completed, submit your assignment using the link marked `Assignment 3: Submission' on KEATS.

You must complete the plagiarism and collusion training before submitting this assignment.

You must also submit complete documentation of your solution. You will find a sample piece of documentation in the Support section on KEATS marked `Sample Assignment Documentation'. Submit your documentation using the link marked `Assignment 3: Documentation Submission' on KEATS.

Students who do not submit documentation along with their code, or vice-versa, will receive a mark of zero.

Any submitted code or documentation that is found to be unduly similar to the code or documentation submitted by any other student(s), will result in a penalty for those involved.

Provisional marks for your code will be released on KEATS within one week of submission. Final assignment grades will be submitted to the exam board at the end of the semester, and will take into consideration the quality of your documentation and the quality of the comments written into your code directly.

For all other queries, see the Support section on KEATS, specifically the document marked `Introduction'.