

# OBJECT ORIENTED PROGRAMMING

[THURSDAY SEPTEMBER 21, 2023: 08:45 AM – 12:00 NOON]

ASSIGNMENTS – 05

CODE: ASSIGN05

NOTES:

- i) Create a main class with the following file naming conventions: If your roll number ends with **abc**, year of admission is 2022 and assignment code is **Assign05** then, use the wrapper file name as follows: **Assign052022abc.java** (use appropriate extension .java suitably).  
*For example, if the roll number ends with 127; year of admission is 2022 & the assignment code is Assign05, then the file name should be **Assign052023127.java***
- ii) Strictly follow the file naming convention. Otherwise, it would attract a penalty up to 20%.

## PROBLEM:

package name: **iiits.oop.m2023**

Choose Java to solve the given problems using nested classes and use random numbers as necessary.

- a) Define a class: **Vehicle** consisting of the following States and Behaviours:

**States: Use suitable data structures for these member variables**

- i. **make**: brand of the car – enum(Tata, Ford, BMW, Rolls-Royce, Benz, Leyland, Isuzu)
- ii. **color**: color of the car – enum(Blue, Black, White, Grey, Silver)
- iii. **year**: the year in which the vehicle has been manufactured in  $\in [2010, 2021]$
- iv. **fuel**: either diesel (1) or petrol (2) or hybrid (3) variants
- v. **engine**: this shows the capacity of the engines in real numbers  $\in [1.0, 3.0]$
- vi. **capacity**: this shows the carrying capacity of the vehicles  $\in [0.0, 15.0]$
- vii. **star**: this shows the number of stars in NCAP crash test  $\in [1, 5]$

**Behaviour: Define different Behaviours**

- i. **isClimbUpHill()**: The capacity of the vehicle to climb the hill (True, False)
- ii. **getAccelerate()**: This shows the acceleration capacity. This may use engine and fuel type as arguments. Acceleration can be defined as the ratio between the change in velocity over change in time. You can assume velocity as a function of using fuel type and engine capacity. For example, if  $v = 9 + 8t - t^2$ , Instance Acceleration  $a = dv/dt = 8 - 2t$  where  $t$  is the time in sec.
- iii. **getNumSeats()**: this is to specify the number of seats one can use to configure the vehicle.
- iv. **getNCAPStars()**: This is to specify the number of stars obtained in NCAP crash test

Vehicle class is always the outer class and any other class defined can be an inner class of any of the following types: Static Nested or Member Inner or Local Inner or Anonymous Inner class

- b) Create a static nested class namely **VehicleData** and write a method inside this inner class to populate an array of **n Vehicles** where **n** an argument  $\in [5, 20]$  with the following state variables. Use random number generation suitably.
- c) Create a Member Inner class namely **Truck** and write a suitable method to generate an array of **k Trucks**, each has a minimum of 2 doors, 2 seating capacity and 12 tons load capacity. Add extra features as you may require for this class. Truck will have more than 4 tyres (and you may consider this as an even number). Use access mechanism in the main method in such a way that this method is called from the scope of outer class. Also write a method to print these details and this method has to be called from the scope of within the class.
- d) Write a method namely **getOdometry()** and inside this method, define a local inner class namely **Speed** that calculates velocity and acceleration given time as an argument. This method should print these two values using a method defined within the local inner class.
- e) Write an interface **PrintDetails** having two methods **PrintData** and **PrintBehaviours**. Implement these two methods from the member inner class to print the attribute-values of the vehicles (using **PrintData**) and the vehicle Behaviors (using **PrintBehaviours**).