OOP's

Monday, 31 July 2023 6:12 PM

Truck Scenario

You are given two incomplete classes Car and Truck.
A Truck is a Car but with the additional behaviors.
Inherit the Car Class into Truck class and build the additional features.
Go through the comments in the prefilled code to implement the Car and Truck classes with the described attributes and methods.

Points to Note:

- The output of the testcase Checking Default Tests is given by the "default_test" function in the prefilled code.
- This coding question does not have the usual input/output testcases. The class defined by you will be tested internally whether the attributes are present or not. So in testcases results you will be shown the rough description of the tests that will be verified.
- You can copy the implementation of Car class from the previous set and add new features on top of that code

Prefilled Code

```
# Truck class should have the following attributes & methods
#
# Old Attributes:
#   color, max_speed, acceleration, tyre_friction, is_engine_started, current_speed
# New Attributes:
#    max_cargo_weight, load
#
# Old Methods:
#   start_engine, stop_engine, accelerate, apply_brakes, sound_horn
#
# Override Methods:
#   sound_horn:
#      - Print "Honk Honk" if truck engine is on
#      - Print "Car has not started yet" if truck engine is off
```

```
#
# New Methods:
#   load_cargo:
#      - This method will have an argument cargo_weight, denoting the weight to
be loaded in the truck.
#      - Truck can load some cargo within max_cargo_weight
#      - When this method is called when the car engine is off, the  current_load of
the truck
#        should increase according to the  cargo_weight  passed as an argument to
this method.
#      - When this method is called when the car engine is on, print the message
"Cannot load cargo during motion"
#      - When the cargo_weight is more than max_cargo_weight,
#          print the message  "Cannot load cargo more than max limit:
{max_cargo_weight}"
#   unload_cargo:
#      - This method will have an argument cargo_weight, denoting the weight to
be unloaded from the truck.
#      - Truck can unload amount of cargo_weight passed as an argument, only
when the truck engine is off.
#      - If the truck engine is on, print the message "Cannot unload cargo during
motion"
#      - Truck load can never go behind 0
#
# When a new Truck is created, the engine should be off by default and
current_speed, load should be 0


# Implement the Car and Truck class appropriately
# Inherit the Car class into Truck class and override the methods which have extra
features

-----------------------------------------------
```