ENCAPSULATION

You will develop a class that will be used to simulate simple car and its motion. This class name will be **Car** and will be expected to have below fields and methods.

Fields
serialNumber: This field must be 5-digit integer. It must be given automatically to each created Car object starting from 10001 and it can never be changed by the Car object user. It can be read by the object user. It can not be accessed directly from other classes.
numberOfDoors: This field can be two or four. It can be modified and read by the object user. If the object user attempts to set other numbers than two or four it must be set to default value which is four. It can not be accessed directly from other classes.
color: This field can be black, white, or red. It can be modified and read by the object user. If the object user attempts to set other colors than black, white, or red it must be set to default color which is white. This color field must be stored in uppercase letters even if the object user attempts to set in lowercase. It can not be accessed directly from other classes.
modelName: This field can be Tiger or Lion It can be modified and read by the object user. If the object user attempts to set other model names than Tiger or Lion it must be set to default model name which is Tiger. This modelName field must be stored in uppercase letters even if the object user attempts to set in lowercase. It can not be accessed directly from other classes.

Methods
start: This method should start the engine and it should print "Engine is started. Ready To Go"
pompGas: This method should provide pomping gas when **move** method is called. It just print "Gas is being pomped". The object user can not call this method directly.
rotateAxle: This method should provide rotating axle when **move** method is called. It just print "Axle is being rotated". The object user can not call this method directly
move: This method executes whole moving process by calling pompGas and rotateAxle methods consecutively. It should print "Car is moving". This method should be accessed from other classes
pompHydraulic: This method should provide pomping hydraulic when **brake** method is called. It just print "Hydraulic is being pomped". The object user can not call this method directly.
movePiston: This method should provide moving piston when **brake** method is called. It just print "Piston is being moved". The object user can not call this method directly.
brake: This method executes whole braking process by calling pompHydraulic and movePiston methods consecutively. It should print "Car is being stopped". This method should be accessed from other classes
stop: This method should stop the engine and it should print "Engine is being stopped"

Also there should be a logical order of calling methods.

1. If engine is already started you should not call start again. Otherwise it will give message "Engine is already started. Ready to go ".
2. To call move engine is started first otherwise it will give message "Engine should be started first" and if car is already moving it will give message "Car is already moving"
3. To call brake, car should be moving otherwise it will give message "Car is not moving"
4. To call stop, engine should be started otherwise it will give message "Engine is already stopped" and Car should not be moving otherwise it will give message "You should brake first"

Constructors
There should be only one constructor with the following signature
Car(int numberOfDoors, String color, String modelName)