Vehicle Rental Management System

Description : Create a simple oop program which simulates a Vehicle Rental Management System by creating the below required classes and interface.

Interfaces :
   1) **IInfoDisplay**

Type	Name	Access Modifier	Return Type	Parameters	Description
Method	ShowInfo()	Public	Void	-	A method which displays vehicle info.

Classes :
   1) **Vehicle**

Type	Name	Access Modifier	Return Type	Parameters	Description
decimal	BasePrice	Protected	-	-	The base hourly price for any vehicle has **a default value of 20 dollars.**
int	VehicleCount	Public static	-	-	A static field which **increases by 1** every time a new vehicle gets rented.
string	OwnerName	Public	-	-	The owner of the rented vehicle
string	ModelName	Public	-	-	The model name of the vehicle.
int	Hours	Public	-	-	Number of hours the vehicle is rented for
constructor	Vehicle	public	-	OwnerName, ModelName, Hours	Initiate the values of the properties with their respective parameter value, and also **increases VehicleCount by 1**
Method	CalculateTotal Price()	Public virtual	decimal	-	Calculates and returns the total price for the vehicle by this formula **(BasePrice * Hours)**

2) SUV : Extends **Vehicle** Class, Implements **IInfoDisplay** interface**.**

Type	Name	Access Modifier	Return Type	Parameters	Description
constructor	SUV	public	-	OwnerName, ModelName, Hours	**Uses base constructor to initiate properties.**
method	CalculateTotal Price()	Public override	decimal	-	**Overrides the base class method** by calculating and returning the total price for the SUV by this formula **((BasePrice + 10) * Hours)**
method	ShowInfo()	Public	-	-	Implements the **IInfoDisplay** interface method which displays the full info about the rented SUV in this format : [SUV] Owner: {OwnerName}, Model: {Model}, Hours: {Hours}, Fee: {CalculateTotalPrice()}

3) MotorCycle : Extends **Vehicle** Class, Implements **IInfoDisplay** interface**.**

Type	Name	Access Modifier	Return Type	Parameters	Description
constructor	MotorCycle	public	-	OwnerName, ModelName, Hours	**Uses base constructor to initiate properties.**
method	CalculateTotal Price()	Public override	decimal	-	**Overrides the base class method** by calculating and returning the total price for the MC by this formula **((BasePrice - 5) * Hours)**
method	ShowInfo()	Public	-	-	Implements the **IInfoDisplay** interface method which displays the full info about the rented MC in this format : [MC] Owner: {OwnerName}, Model: {Model}, Hours: {Hours}, Fee:{CalculateTotalPrice()}

Main Method: In Main Method create an object of each class and use their methods to calculate prices and show info, and at the end display the number of rented vehicles.