C# Assignments on Abstract Class, Interface and Partial Class

Assignment 1. Abstract Class

Create an abstract class Vehicle that has an abstract method StartEngine() and a concrete method StopEngine(). Create derived classes Car and Motorcycle that implement the StartEngine() method and override it to show specific behavior for each type of vehicle.

Assignment 2. Virtual Functions

Create a base class Animal with a virtual method MakeSound(). Derive classes Dog and Cat that override the MakeSound() method to provide their specific implementation.

Assignment 3. Interface

Create an interface IDrive with a method Drive(). Implement this interface in a Car and Truck class, with each class having its own implementation of Drive().

Assignment 4. Interface vs. Abstract Class

Write a program that demonstrates the difference between an abstract class and an interface by creating an abstract class Bird with an abstract method Fly(), and an interface ISwim with a method Swim().

Assignment 5. Static Class

Create a static class MathOperations with a static method Add() and Multiply(). Demonstrate calling these methods without creating an instance of the class.

Assignment 6. Extension Methods

Create an extension method IsEven() for the int type that returns true if the number is even and false if it is odd.

Assignment 7. Partial Class

Create a partial class Person that is defined in two files. One file should have the property Name, and the other file should have the method ShowDetails().

Assignment 8. Partial Methods

Create a partial class Employee with a partial method CalculateSalary(). Implement the partial method in another part of the class and demonstrate its usage.

Assignment 9. Indexer

Create a Library class that contains an array of Book objects. Implement an indexer that allows accessing the books by index. Write a method to display all the books in the library.

Assignment 10. Exception Handling

Write a method Divide that takes two integers as input and returns their division. If a division by zero occurs, catch the exception and display a custom error message. Demonstrate exception handling with a try-catch-finally block.

Assignment 11. Enum

Create an enum CarType with values Sedan, SUV, Truck, and Coupe. Write a Car class with a property Type of type CarType. Write a method that takes a CarType value and displays a message specific to that type of car.

Assignment 12. Attributes

Define a custom attribute DeveloperAttribute that takes the name of the developer and the date when the code was last modified. Apply this attribute to a class Calculator and its method Add. Retrieve and display the attribute information at runtime.