Programming with C#

# Class Inheritance Practice

## Implement Inheritance

1. Open Visual Studio if it is not already open.
2. Navigate to the folder where you installed the companion content.
3. In the Chapter 12 folder, choose either Windows 7 or Windows 8 and open the Vehicles project
4. Add a new class to the project called Vehicles.
5. Add the following code to the Vehicle class:  
     
   **public void StartEngine(string noiseToMakeWhenStarting)   
   {   
    Console.WriteLine("Starting engine: {0}", noiseToMakeWhenStarting);   
   }  
      
   public void StopEngine(string noiseToMakeWhenStopping)   
   {   
    Console.WriteLine("Stopping engine: {0}", noiseToMakeWhenStopping);   
   }**
6. Add a new class to the project called Airplane.
7. Modify the default Airplane class template code so that it inherits from Vehicle.  
     
   class Airplane **: Vehicle**
8. Because an airplane has different behaviors, add the following methods to the Airplane class:  
     
   **public void TakeOff()   
   {   
    Console.WriteLine("Taking off");   
   }   
     
   public void Land()   
   {   
    Console.WriteLine("Landing");   
   }**
9. Add a new class to the project called Car.
10. Make Car inherit from Vehicle.
11. Add the following methods to Car:  
      
    **public void Accelerate()   
    {   
     Console.WriteLine("Accelerating");   
    }   
      
    public void Brake()   
    {   
     Console.WriteLine("Braking");   
    }**
12. Modify the Vehicle class to add a virtual method called Drive:  
      
    **public virtual void Drive()   
    {  
     Console.WriteLine("Default implementation of the Drive method");   
    }**
13. Display the Program.cs file and in the doWork() method, add the following code:  
      
    **Console.WriteLine("Journey by airplane:");   
    Airplane myPlane = new Airplane();   
    myPlane.StartEngine("Contact");   
    myPlane.TakeOff();   
    myPlane.Drive();   
    myPlane.Land();   
    myPlane.StopEngine("Whirr");  
      
    Console.WriteLine("\nJourney by car:");   
    Car myCar = new Car();   
    myCar.StartEngine("Brm brm");   
    myCar.Accelerate();   
    myCar.Drive();   
    myCar.Brake();   
    myCar.StopEngine("Phut phut");**
14. Press CTRL + F5 to start the application without debugging.
15. Fix any errors that may occur.
16. Verify the output in the console window and then close the application.
17. Modify the Airplane class and override the Drive method:  
      
    **public override void Drive()   
    {   
     Console.WriteLine("Flying");   
    }**
18. Also override the Drive method in the Car class:  
      
    **public override void Drive()   
    {   
     Console.WriteLine("Motoring");   
    }**
19. Press CTRL + F5 to run the application and verify the output has changed.
20. Press Enter to stop the program and return to Visual Studio
21. Go back to Program.cs and add this code to the bottom of the doWork() method to demonstrate polymorhpism:  
      
    **Console.WriteLine("\nTesting polymorphism");   
    Vehicle v = myCar;   
    v.Drive();   
    v = myPlane;   
    v.Drive();**
22. Press CTRL + F5 and note the output. This demonstrates how a base class can be assigned a sub class in its hierarchy.
23. Press Enter to close the application and return to Visual Studio.
24. Close Visual Studio 2013.