C# Programming

Notes by JACPro

C# 1.x Notes

Classes

public class Customer

{

//fields

//properties

//methods

//events

}

Customer object2;

Declares type and namespace but will be null unless an object is assigned.

Customer object2 = new Customer();

Create and assign an object.

Classes are reference types.

Customer object3 = object2;

object2 and object3 refer to the same object, so changes to one will be reflected in the other.

Inheritance

You can only inherit from a single class.

public class Manager : Employee

{

//fields, properties, methods and events inherited from Employee

//new fields, properties, methods and events for Manager

}

Abstract Classes

The purpose of an abstract class is to provide a common definition of a base class that multiple derived classes can share.

Abstract classes cannot be instantiated.

public abstract class A

{

//Class members here

}

Sealed Classes

Sealed classes cannot be used as base classes because they prevent derivation.

public sealed class D

{

//Class members here

}

A method, index, property or event can be declared as sealed in a derived class when overwriting a virtual member of the base class; this prevents the virtual aspect from being derived further.

public class D : C

{

public sealed override void DoWork() { }

}

Auto-Implementation

Auto-implementation is used when no additional logic is required in the property accessors.

They are used to make private variables visible outside the class by using properties with get; set; methods to expose private fields.