[.NET(C#)]
attribute翻译成特性，用来标识类，方法 。
property翻译为属性，性质用于存取类的字段 。
markup 翻译成标记。
tag 翻译成标签。

[.NET(C#)]
程序集的一个重要特性是它们包含的元数据描述了对应代码中定义的类型和方法。

[.NET(C#)]
ASP 页面有时显示比较慢，因为服务器端代码是解释性的不是编译的。
ASP.NET 页面是结构化的。每个页面都是一个继承了.NET 类 System.Web.UI.Page 的类。

[.NET(C#)]
重写override：是指子类重新定义父类的虚函数的做法。
重载overload：是指允许存在多个同名函数，而函数签名不同（参数表不同：或许参数个数不同，或许参数类型不同，或许两者都不同）。重载的概念并不属于“面向对象编程”。

[.NET(C#)]
ref 关键字使参数按引用传递。其效果是，当控制权传递回调用方法时，在方法中对参数所做的任何更改都将反映在该变量中。若要使用 ref 参数，则方法定义和调用方法都必须显式使用 ref 关键字。
out 关键字会导致参数通过引用来传递。这与 ref 关键字类似，不同之处在于 ref 要求变量必须在传递之前进行初始化。若要使用 out 参数，方法定义和调用方法都必须显式使用 out 关键字。

[.NET(C#)]
ADO和ADO.NET的区别:
ADO使用OLE DB接口并基于微软的COM技术；而ADO.NET拥有自己的ADO.NET接口并且基于微软的.NET体系架构。
ADO以Recordset存储，而ADO.NET则以DataSet表示。
Recordset看起来更像单表，如果让Recordset以多表的方式表示就必须在SQL中进行多表连接。反之，DataSet可以是多个表的集合。
ADO 的运作是一种在线方式，这意味着不论是浏览或更新数据都必须是实时的。
ADO.NET则使用离线方式，在访问数据的时候ADO.NET会利用XML制作数据的一份幅本
ADO.NET的数据库连接也只有在这段时间需要在线。

[.NET(C#)]
new 关键字用法：
1) new 运算符 ：用于创建对象和调用构造函数。
2) new 修饰符 ：用于向基类成员隐藏继承成员。
3) new 约束 ：用于在泛型声明中约束可能用作类型参数的参数的类型。

[.NET(C#)]
C#中，string str = null 与 string str = ""，说明区别：
string str = "" 初始化对象分配空间。
string str = null 表示一个空引用,没有占用空间。

[.NET(C#)]
ADO.NET相对于ADO等主要有什么改进？
1) ado.NET不依赖于ole db提供程序,而是使用.NET托管提供的程序。
2) 不使用com。
3) 不在支持动态游标和服务器端游。
4) 可以断开connection而保留当前数据集可用。
5) 强类型转换。
6) xml支持。

[.NET(C#)]
DataGrid的Datasouse可以连接什么数据源：
1) DataTable
2) DataView
3) DataSet
4) DataViewManager
5) 任何实现IListSource接口的组件
6) 任何实现IList接口的组件

[.NET(C#)]
反射：
可以使用反射动态地创建类型的实例，将类型绑定到现有对象，或从现有对象中获取类型。然后，可以调用类型的方法或访问其字段和属性。

[.NET(C#)]
序列化：
序列化是将对象状态转换为可保持或传输的格式的过程。
与序列化相对的是反序列化，它将流转换为对象。这两个过程结合起来，可以轻松地存储和传输数据。

[.NET(C#)]
可访问性级别有哪几种：
1) public 访问不受限制。
2) protected 在它的类中可访问并且可由派生类访问。。
3) internal 只有在同一程序集的文件中，内部类型或成员才是可访问的。
4) protected internal 同一个程序集中的所有类，以及所有程序集中的子类都可以访问。
5) private 私有成员只有在声明它们的类和结构体中才是可访问的。

[.NET(C#)]

O/R Mapping 的原理:利用反射，配置将对象和数据库表映射。

[.NET(C#)]

sealed 修饰符有什么特点：

1) sealed 修饰符可以应用于类、实例方法和属性。密封类不能被继承。密封方法会重写基类中的方法，但其本身不能在任何派生类中进一步重写。当应用于方法或属性时，sealed 修饰符必须始终与 override一起使用。
2) 将密封类用作基类或将 abstract 修饰符与密封类一起使用是错误的。
3) 结构是隐式密封的；因此它们不能被继承。

[.NET(C#)]

详述.NET里class和struct的异同：

相同点：

1) 语法类似。

不同点：

1) class是引用类型，继承自System.Object类; struct是值类型，继承自System.ValueType类，因此不具多态性。但是注意，System.ValueType是个引用类型。
2) 从职能观点来看，class表现为行为; 而struct常用于存储数据。
3) class支持继承，可以继承自类和接口; 而struct没有继承性，struct不能从class继承，也不能作为class的基类，但struct支持接口继承。
4) 实例化时，class要使用new关键字; 而struct可以不使用new关键字，struct在声明时就进行了初始化过程，所有的成员变量均默认为0或null。

[.NET(C#)]

如何选择结构还是类

1) 堆栈的空间有限，对于大量的逻辑的对象，创建类要比创建结构好一些。
2) 结构表示如点、矩形和颜色这样的轻量对象。例如，如果声明一个含有 1000 个点对象的数组，则将为引用每个对象分配附加的内存。在此情况下，结构的成本较低。
3) 在表现抽象和多级别的对象层次时，类是最好的选择。
4) 大多数情况下该类型只是一些数据时，结构时最佳的选择。

[.NET(C#)]

抽象类（abstract class）和接口（interface）的区别：

抽象类

1) 抽象方法只作声明，而不包含实现，可以看成是没有实现体的虚方法。
2) 抽象类不能被实例化。
3) 抽象类可以但不是必须有抽象属性和抽象方法，但是一旦有了抽象方法，就一定要把这个类声明为抽象类。
4) 具体派生类必须覆盖基类的抽象方法。
5) 抽象派生类可以覆盖基类的抽象方法，也可以不覆盖。如果不覆盖，则其具体派生类必须覆盖它们。

[.NET(C#)]

接口

1) 接口不能被实例化。
2) 接口只能包含方法声明。
3) 接口的成员包括方法、属性、索引器、事件。
4) 接口中不能包含常量、字段(域)、构造函数、析构函数、静态成员。
5) 接口中的所有成员默认为public，因此接口中不能有private修饰符。
6) 派生类必须实现接口的所有成员。
7) 一个类可以直接实现多个接口，接口之间用逗号隔开。
8) 一个接口可以有多个父接口，实现该接口的类必须实现所有父接口中的所有成员。

[.NET(C#)]

抽象类和接口的异同：

相同点：

1) 都可以被继承。
2) 都不能被实例化。
3) 都可以包含方法声明。
4) 派生类必须实现未实现的方法。

区 别：

1) 抽象基类可以定义字段、属性、方法实现。接口只能定义属性、索引器、事件、和方法声明，不能包含字段。
2) 抽象类是一个不完整的类，需要进一步细化，而接口是一个行为规范。微软的自定义接口总是后带able字段，证明其是表述一类“我能做。。。”。
3) 接口可以被多重实现，抽象类只能被单一继承。
4) 抽象类更多的是定义在一系列紧密相关的类间,而接口大多数是关系疏松但都实现某一功能的类中。
5) 抽象类是从一系列相关对象中抽象出来的概念， 因此反映的是事物的内部共性；接口是为了满足外部调用而定义的一个功能约定， 因此反映的是事物的外部特性。
6) 接口基本上不具备继承的任何具体特点,它仅仅承诺了能够调用的方法。
7) 接口可以用于支持回调,而继承并不具备这个特点。
8) 抽象类实现的具体方法默认为虚的，但实现接口的类中的接口方法却默认为非虚的，当然您也可以声明为虚的。
9) 如果抽象类实现接口，则可以把接口中方法映射到抽象类中作为抽象方法而不必实现，而在抽象类的子类中实现接口中方法。

[.NET(C#)]

什么叫应用程序域：

1) 操作系统和运行库环境通常会在应用程序间提供某种形式的隔离。
2) 应用程序域为安全性、可靠性、版本控制以及卸载程序集提供了隔离边界。
3) 应用程序域可以理解为一种轻量级进程。起到安全的作用。占用资源小。

[.NET(C#)]

强类型：

为所有变量指定数据类型称为“强类型”。C#是强类型语言。

[.NET(C#)]
装箱和拆箱：
1) 从值类型接口转换到引用类型：装箱。
2) 从引用类型转换到值类型：拆箱。

[.NET(C#)]
托管代码：
使用基于公共语言运行库的语言编译器开发的代码称为托管代码；托管代码具有许多优点，例如：跨语言集成、跨语言异常处理、增强的安全性、版本控制和部署支持、简化的组件交互模型、调试和分析服务等。

[.NET(C#)]
CTS：通用系统类型 Common Type System。
所有.NET语言共享这一类型系统，实现它们之间无缝的互操作。该方案还提供了语言之间的继承性。

[.NET(C#)]
CLR：公共语言运行库 Common Language Runtime。
是一个运行时环境，它负责资源管理（内存分配和垃圾收集），并保证应用和底层操作系统之间必要的分离。

[.NET(C#)]
CLS：公共语言规范 Common Language Specification
可以保证C#组件与其他语言组件间的互操作性。

[.NET(C#)]
委托：
1) 委托是一种引用方法的类型。
2) 委托类似于 C++ 函数指针，但它是类型安全的。
3) 委托允许将方法作为参数进行传递。
4) 委托可用于定义回调方法。

[.NET(C#)]
活动目录的作用：
1) Active Directory存储了有关网络对象的信息，并且让管理员和用户能够轻松地查找和使用这些信息。
2) Active Directory 使用了一种结构化的数据存储方式，并以此作为基础对目录信息进行合乎逻辑的分层组织。

[.NET(C#)]
在.NET 中，配件的意思：
程序集（中间语言，源数据，资源，装配清单）。

[.NET(C#)]

UDDI、WSDL 的意义及其作用：

1) UDDI（Universal Description, Discovery and Integration）：通用描述、发现与集成服务；是一种目录服务，企业可以使用它对 Web services 进行注册和搜索。
2) WSDL（Web Services Description Language）：是一个用来描述 Web 服务和说明如何与 Web 服务通信的 XML 语言。

[.NET(C#)]

值类型和引用类型的区别：

1) 值类型通常被分配在栈上，它的变量直接包含变量的实例，使用效率比较高。
2) 引用类型分配在托管堆上，引用类型的变量通常包含一个指向实例的指针，变量通过该指针来引用实例。
3) 一个是值COPY，一个是地址COPY。

	值类型	引用类型
内存分配地点	分配在栈中	分配在堆中
效率	效率高，不需要地址转换	效率低，需要进行地址转换
内存回收	使用完后，立即回收	使用完后，不是立即回收，等待GC回收
赋值操作	进行复制，创建一个同值新对象	只是对原有对象的引用
函数参数与返回值	是对象的复制	是原有对象的引用，并不产生新的对象
类型扩展	不易扩展	容易扩展，方便与类型扩展

[.NET(C#)]

ASP.NET的身份验证方式：

Windows 身份验证提供程序：

提供有关如何将 Windows 身份验证与 Microsoft Internet 信息服务 (IIS) 身份验证，结合使用来确保 ASP.NET 应用程序安全的信息。

Forms 身份验证提供程序：

提供有关如何使用您自己的代码创建应用程序特定的登录窗体并执行身份验证的信息。

使用 Forms 身份验证的一种简便方法是使用 ASP.NET 成员资格和 ASP.NET 登录控件，它们一起提供了一种只需少量或无需代码就可以收集、验证和管理用户凭据的方法。

Passport 身份验证提供程序

提供有关由 Microsoft 提供的集中身份验证服务的信息，该服务为成员站点提供单一登录和核心配置。

[.NET(C#)]

GC 是什么? 为什么要有 GC?

GC是控制系统垃圾回收器（一种自动回收未使用内存的服务）。

当使用可用内存不能满足内存请求时，垃圾回收会自动进行。

调用方法：GC.Collect();

[.NET(C#)]

SOAP 及其应用：

SOAP（Simple Object Access Protocol ）简单对象访问协议。

1) 是一种轻量的、简单的、基于 XML 的协议，它被设计成在 WEB 上交换结构化的和固化的信息。
2) SOAP 可以和现存的许多因特网协议和格式结合使用，包括超文本传输协议（ HTTP），简单邮件传输协议（SMTP），多用途网际邮件扩充协议（MIME）。
3) 它还支持从消息系统到远程过程调用（RPC）等大量的应用程序。

[.NET(C#)]

.NET 中的垃圾回收机制：

.NET Framework 的垃圾回收器管理应用程序的内存分配和释放。每次您使用 new 运算符创建对象时，运行库都从托管堆为该对象分配内存。只要托管堆中有地址空间可用，运行库就会继续为新对象分配空间。但是，内存不是无限大的。最终，垃圾回收器必须执行回收以释放一些内存。垃圾回收器优化引擎根据正在进行的分配情况确定执行回收的最佳时间。当垃圾回收器执行回收时，它检查托管堆中不再被应用程序使用的对象并执行必要的操作来回收它们占用的内存。

[.NET(C#)]

ASP.NET 页面之间传递值的几种方式，与各自的优缺点：

1) 使用 QueryString 变量：Response.Redirect(字符串)；简单，显示于地址栏，长度有限。
2) 使用 Session 变量：简单；但易丢失。
3) 使用 Application 变量：全局；易被误操做。
4) 使用 Server.Transfer：可以传递各种数据类型的值；显示在地址栏。
5) 使用 Cookie 变量：使用简单；存在客户端可能被伪造。

[.NET(C#)]

索引器：

索引器允许类或结构的实例按照与数组相同的方式进行索引。索引器类似于属性，不同之处在于它们的访问器采用参数。可以用任意类型索引。

1) 索引器使得对象可按照与数组相似的方法进行索引。
2) get 访问器返回值。set 访问器分配值。
3) this 关键字用于定义索引器。
4) value 关键字用于定义由 set 索引器分配的值。
5) 索引器不必根据整数值进行索引，由您决定如何定义特定的查找机制。
6) 索引器可被重载。
7) 索引器可以有多个形参，例如当访问二维数组时。

[.NET(C#)]

进程和线程的区别：

1) 进程是系统进行资源分配和调度的单位。
2) 线程是CPU调度和分派的单位。
3) 一个进程可以有多个线程，这些线程共享这个进程的资源。

[.NET(C#)]

ADO.NET 的组成部分：

ADO.NET 有两个重要组成部分，即 DataSet 和.NET 数据提供者。

.NET 数据提供者的对象包括：Connection、Command、CommandBuilder、DataReader 和 DataAdapter。

1) Connection 对象：主要是开启程序和数据库之间的连结。
2) Command 对象：主要可以用来对数据库发出一些指令；这个对象是架构在 Connection 对象上的。
3) DataAdapter 对象：主要是在数据源以及 DataSet 之间执行数据传输的工作；这个对象是架构在 Command 对象上，并提供了许多配合 DataSet 使用的功能。
4) DataSet 对象：可以视为一个暂存区（Cache），可以把从数据库中所查询到的数据保留起来；DataSet 对象可以说是 ADO.NET 中重量级的对象，这个对象架构在 DataAdapter 对象上，本身不具备和数据源沟通的能力；也就是说我们是将 DataAdapter 对象当做 DataSet 对象以及数据源间传输数据的桥梁。
5) DataReader 对象只是一次一笔向下循序的读取数据源中的数据，而且这些数据是只读的，并不允许作其它的操作。

[.NET(C#)]

常用的调用 WebService 的方法：

1) 使用WSDL.exe命令行工具。
2) 使用 VS.NET 中的 Add Web Reference 菜单选项。

[.NET(C#)]

软.NET 构架下 Remoting 和 webservice 两项技术的理解以及实际中的应用：

Remoting：

1) .NET Remoting 的工作原理是：服务器端向客户端发送一个进程编号，一个程序域编号，以确定对象的位置。
2) Remoting 是.NET 中用来跨越 machine,process,appdomain 进行方法调用的技术。
3) 对于三成结构的程序，就可以使用 Remoting 技术来构建。
4) 它是分布应用的基础技术.相当于以前的 DCOM 。

Web Service:

1) Web Service 是一种构建应用程序的普通模型。
2) 并能在所有支持 internet 网通讯的操作系统上实施。
3) Web Service 令基于组件的开发和 web 的结合达到最佳。

WS 主要是可利用 HTTP，穿透防火墙。而 Remoting 可以利用 TCP/IP，二进制传送提高效率。

[.NET(C#)]

启动一个线程是用 run()还是 start()？

启动一个线程是调用 start()方法，导致操作系统将当前实例的状态更改为 ThreadState.Running。

[.NET(C#)]
构造器（构造函数、构造方法）Constructor 是否可被 override（重写）？
构造器 Constructor 不能被继承，因此不能重写 override，但可以被重载 Overloade。

[.NET(C#)]
静态成员 override、virtual 或 abstract。
抽象类不能是密封的 sealed 或静态的 static。

[.NET(C#,J#)]
final, finally, finalize 的区别：

final(J#, C#见Sealed):
1) final修饰符来限定变量、字段、方法和类。
2) 如果一个变量被声明为 final，则该变量只能设置一次。
3) final 字段可由它的类构造函数设置一次。
4) 使用 final 修饰符声明的方法不能被重写或隐藏。
5) Final 类不能被继承或扩展。因此 一个类不能既被声明为 abstract的，又被声明为 final的。

finally :
finally 块用于清除 try 块中分配的任何资源，以及运行任何即使在发生异常时也必须执行的代码。控制总是传递给 finally 块，与 try 块的退出方式无关。

finalize:
Object.Finalize 方法，允许 Object 在“垃圾回收”回收 Object 之前尝试释放资源并执行其他清理操作。

[.NET(C#)]
堆和栈的区别：
1) 栈：由编译器自动分配、释放。在函数体中定义的变量通常在栈上。存放值类型。
2) 堆：一般由程序员分配释放。用 new 等分配内存函数分配得到的就是在堆上。存放引用类型。

[.NET(C#)]
在 c#中 using 和 new 这两个关键字有什么意义：
using 关键字有两个主要用途：
1) 作为指令，用于为命名空间创建别名或导入其他命名空间中定义的类型。
2) 作为语句，用于定义一个范围，在此范围的末尾将释放对象。
new 关键字：新建实例或者隐藏父类方法

[.NET(C#)]
概述 XML
即可扩展标记语言
1) eXtensible Markup Language.标记是指计算机所能理解的信息符号
2) 通过此种标记，计算机之间可以处理包含各种信息的文章等。

[.NET(C#)]
什么是 code-Behind 技术：
就是页面与代码分离；ASPX,RESX 和 CS 三个后缀的文件，这个就是代码分离。
实现了 HTML 代码和服务器代码分离，方便代码编写和整理。

[.NET(C#)]
XML 与 HTML 的主要区别：
1) XML是区分大小写字母的，HTML不区分。
2) 在HTML中，如果上下文清楚地显示出段落或者列表键在何处结尾，那么你可以省略</p>或者
之类的结束标记；在XML中，绝对不能省略掉结束标记。
3) 在XML中，拥有单个标记而没有匹配的结束标记的元素必须用一个 / 字符作为结尾。这样分析器就知道不用 查找结束标记了。
4) 在XML中，属性值必须分装在引号中。在HTML中，引号是可用可不用的。
5) 在 HTML 中，可以拥有不带值的属性名。在 XML 中，所有的属性都必须带有相应的值。

[.NET(C#)]
.NET 的错误处理机制：
.NET错误处理机制采用try->catch->finally结构。
发生错误时，层层上抛，直到找到匹配的 catch 为止。

[.NET(C#，JAVA)]
Static Nested Class（嵌套类） 和 Inner Class（内部类）的不同:
Static Nested Class是被声明为静态（static）的内部类，它可以不依赖于外部类实例被实例化。
而通常的内部类需要在外部类实例化后才能实例化。

[.NET(C#)]
error 和 exception 有什么区别：
error 表示恢复不是不可能但很困难的情况下的一种严重问题。比如说内存溢出。
不可能指望程序能处理这样的情况。
exception 表示一种设计或实现问题。
也就是说，它表示如果程序运行正常，从不会发生的情况。

[.NET(C#)]
UDP连接和TCP连接的异同:
1) TCP（Transmission Control Protocol）传输控制协议：一种面向连接的、可靠的、基于字节流的运输层通信协议，三次握手。
2) UDP（User Datagram Protocol）用户数据报协议：它不属于连接型协议，因而具有资源消耗小，处理速度快的优点。缺点是易丢失数据包。

[.NET(C#)]
C#中所有对象共同的基类是:System.Object。

[.NET(C#)]
System.String 和System.StringBuilder有什么区别？
1) System.String是不可变的字符串。
2) System.StringBuilder存放了一个可变的字符串，并提供一些对这个字符串修改的方法。
3) String类在执行字符串拼接的操作上，用“+”会产生新的对象，占用内存。
4) StringBuilder类只是修改字符串的内容，不建立新的对象。

[.NET(C#)]
const 和 readonly 有什么区别？
1) const 字段只能在该字段的声明中初始化。
2) 不允许在常数声明中使用 static 修饰符。
3) readonly 字段可以在声明或构造函数中初始化。因此，根据所使用的构造函数，readonly 字段可能具有不同的值。
4) 另外，const 字段是编译时常数，而 readonly 字段可用于运行时常数。

[.NET(C#)]
C#中的委托是什么？事件是不是一种委托？
委托可以把一个方法作为参数代入另一个方法。
委托可以理解为指向一个函数的引用。
事件是一种特殊的委托。

[.NET(C#)]
如果在一个B/S结构的系统中需要传递变量值，但是又不能使用Session、Cookie、Application，您有几种方法进行处理？
this.Server.Transfer。

[.NET(C#)]
用.net做B/S结构的系统，您是用几层结构来开发，每一层之间的关系以及为什么要这样分层？
一般为3层：数据访问层，业务层，表示层。
1) 数据访问层对数据库进行操作（增删查改）。
2) 业务层一般分为二层，业务表观层实现与表示层的沟通，业务规则层实现用户密码的安全等。
3) 表示层为了与用户交互例如用户添加表单。

优点： 分工明确，条理清晰，易于调试，而且具有可扩展性。
缺点： 增加成本。

[.NET(C#)]
什么是受管制的代码？
在类型或成员的声明中使用 unsafe 修饰符。因此，类型或成员的整个正文范围均被视为不安全上下文，无法由 CLR 进行验证的代码。

[.NET(C#)]
什么是强类型系统?
RTTI：类型识别系统。

[.NET(C#)]
面向对象语言三大特性：
封装、继承、多态。

举例说明：
封装：使用属性来访问变量。
继承：父类、子类。
多态：重写。

[.NET(C#)]
能用foreach遍历访问的对象需要实现 _____接口或声明_____方法的类型。
答：IEnumerable 、 GetEnumerator。

[.NET(C#)]
接口是否可继承接口? 抽象类是否可实现(implements)接口? 抽象类是否可继承实体类(concrete class)?
接口可以继承接口。抽象类可以实现(implements)接口，抽象类是否可继承实体类，但前提是实体类必须有明确的构造函数。

[.NET(C#)]
是否可以继承String类?
String类是sealed类故不可以继承。

[.NET(C#)]
数组有没有length()这个方法? String有没有length()这个方法?
数组、String类都没有Length()方法，它们只有Length属性。

[.NET(C#)]
Sleep() 和 Wait() 有什么区别?
sleep()不释放资源，wait()释放资源。
sleep()方法是将当前线程挂起指定的时间。
wait()释放对象上的锁并阻塞当前线程，直到它重新获取该锁。

[.NET(C#)]
Session有什么重大BUG，微软提出了什么方法加以解决?
是IIS中由于有进程回收机制，系统繁忙的话Session会丢失，可以用Sate server或SQL Server数据库的方式存储Session，不过这种方式比较慢，而且无法捕获Session的END事件。

[.NET(C#)]
成员变量和成员函数前加static的作用:
即使没有创建类的实例，也可以调用该类中的静态方法、字段、属性或事件。如果创建了该类的任何实例，不能使用实例来访问静态成员。静态成员通常用于表示不会随对象状态而变化的数据或计算。

[.NET(C#)]
ASP.NET与ASP相比，主要有哪些进步？
答：asp 解释型，aspx 编译型，性能提高，有利于保护源码。

[.NET(C#)]
请指出GAC的含义？
全局程序集缓存。

[.NET(C#)]
向服务器发送请求有几种方式？
get，post。get 方式提交的数据量有限制，理论上 post 没有限制，可传较大量的数据。

[.NET(C#)]
DataReader 与 Dataset 有什么区别？
一个是只能向前的只读游标，一个是内存中的表。

无连接一般用在大数据量访问上，比如你要获得某些历史数据，一般使用 DataSet 来存储数据。
优点：就是通过一次访问就可以取得大量数据，降低网络开销，使用灵活等。
缺点：就是得到的数据有可能不是最新的，在更新的时候需要做校验。

保持连接一般都用在需要快速访问、取得数据量较小的场合下，比如用户登录验证。一般使用 DataReader。
优点：速度快。占用资源小。
缺点：只能用来浏览，不可以做修改。

[.NET(C#)]
软件开发过程一般有几个阶段？每个阶段的作用？
1) 需求分析：确定要做什么。
2) 设计：怎么做。
3) 代码：做。
4) QA & 测试：保证质量、验证。
5) 部署 & 交付：安装、使用。

[.NET(C#)]
用Singleton如何写设计模式：
static属性里面new ,构造函数private。

[.NET(C#)]
什么叫做 SQL 注入，如何防止？请举例说明。
所谓 SQL 注入，就是通过把 SQL 命令插入到 Web 表单递交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的 SQL 命令。

防止：
1) 对用户输入的内容进行校验。
2) 少使用动态拼接 SQL 语句。
3) 过滤关键字，如：单引号。
4) 使用IDataParamater。

[.NET(C#)]
什么是 Application Pool？
应用程序池。应用程序池是将一个或多个应用程序链接到一个或多个工作进程集合的配置。

[.NET(C#)]
virtual 修饰符不能与 static、abstract 和 override 修饰符一起使用。

[.NET(C#)]
什么是虚函数？什么是抽象函数？
1) 虚函数：没有实现的，可由子类继承并重写的函数。
2) 抽象函数：规定其非虚子类必须实现的函数，必须被重写。

[.NET(C#)]
什么是 Web Service？
Web service 就是一个应用程序，它向外界暴露出一个能够通过 Web 进行调用的 API。

[.NET(C#)]
什么是 ASP.net 中的用户控件？
创建自己的自定义的可重用控件，所采用的方法与用来开发 ASP.NET 网页的方法相同。这些控件称作用户控件。

[.NET(C#)]
列举一下你所了解的 XML 技术及其应用：
xml 用于配置，用于保存静态数据类型.接触 XML 最多的是 web Services 和 config。

[.NET(C#)]
c#中的三元运算符是？
？：。格式如下：ndition ？first_expression ： second_expression。

[.NET(C#)]
当整数a赋值给一个object对象时，整数a将会被？
装箱。

[.NET(C#)]
委托声明的关键字是_____?
delegate.

[.NET(C#)]
在ASP.NET户控件都必须继承自_____?
Control。

[.NET(C#)]
在.Net中所有可序列化的类都被标记为_____?
[Serializable]。

[.NET(C#)]
在.Net托管代码中我们不用担心内存漏洞，这是因为有了_____?
GC。

[.NET(C#)]
当类T只声明了私有实例构造函数时，则在T的程序文本外部，_____（可以 or 不可以）从T派生出新的类， ____（可以 or 不可以）直接创建T的任何实例。
答：不可以，不可以

[.NET(C#)]
在.Net中，类System.Web.UI.Page 可以被继承么?
可以。

[.NET(C#)]
利用operator声明且仅声明了==，有什么错误么?
要同时修改 Equale 和 GetHash() ? 重载了"==" 就必须重载 "!="。

[.NET(C#)]
在.net（C# or vb.net）中，Application.Exit() 与Form.Close()有什么不同?
Application.Exit()：关闭所有应用程序窗口。
Form.Close()：关闭窗体。

[.NET(C#)]
62-63=1 等式不成立，请移动一个数字（不可以移动减号和等于号），使得等式成立，如何移动?
62 移动成 2 的 6 次方；2^6。

[.NET(C#)]
C#可否对内存进行直接的操作?
可以使用指针。也就是非安全代码，就是不在 CLR 完全控制下执行的代码，它有可能会导致一些问题，因此他们必须用 “unsafe” 进行表明。

[.NET(C#)]
某一密码仅使用K、L、M、N、O共5个字母，密码中的单词从左向右排列，密码单词必须遵循如下规则：
1) 密码单词的最小长度是两个字母，可以相同，也可以不同。
2) K不可能是单词的第一个字母。
3) 如果L出现，则出现次数不止一次。
4) M不能使最后一个也不能是倒数第二个字母。
5) K出现，则N就一定出现。
6) O如果是最后一个字母，则L一定出现。

问题一：下列哪一个字母可以放在LO中的O后面，形成一个3个字母的密码单词？
A. K
B. L
C. M
D. N
答案：B。

问题二：如果能得到的字母是K、L、M，那么能够形成的两个字母长的密码单词的总数是多少？
A. 1个
B. 3个
C. 6个
D. 9个
答案：A。

问题三：下列哪一个是单词密码？
A. KLLN
B. LOML
C. MLLO
D. NMKO
答案：C。

[.NET(C#)]
&和&&的区别：
&是位运算符，表示按位与运算。
&&是逻辑运算符，表示逻辑与（and）

[.NET(C#，JAVA)]
HashMap 和 Hashtable 的区别：
HashMap 是 Hashtable 的轻量级实现（非线程安全的实现），他们都完成了 Map 接口，主要区别在于 HashMap 允许空(null)键值(key),由于非线程安全，效率上可能高于 Hashtable。

[.NET(C#)]
重载的方法是否可以改变返回值的类型?
可以。

[.NET(C#)]
大概描述一下ASP.NET服务器控件的生命周期：
1. 初始化
2. 加载视图状态
3. 处理回发数据
4. 加载
5. 发送回发更改通知
6. 处理回发事件
7. 预呈现
8. 保存状态
9. 呈现
10. 处置
11. 卸载

[.NET(C#)]
<%# %> 和 <% %> 有什么区别?
<%# %>：数据绑定表达式语法。所有数据绑定表达式都必须包含在 <%# 和 %> 字符之间。
<% %> ：ASP.NET 网页中的嵌入式代码块。

[.NET(C#)]
你觉得 ASP.NET 2.0（VS2005）和你以前使用的开发工具（.Net 1.0 或其他）有什么最大的区别？你在以前的平台上使用的哪些开发思想（pattern / architecture）可以移植到 ASP.NET 2.0 上 (或者已经内嵌在 ASP.NET 2.0 中)?
1. ASP.NET 2.0 把一些代码进行了封装打包,所以相比 1.0 相同功能减少了很多代码。
2. 同时支持代码分离和页面嵌入服务器端代码两种模式,以前 1.0 版本,.NET 提示帮助只有在分离的代码文件,无 法 在页面嵌入服务器端代码获得帮助提示。
3. 代码和设计界面切换的时候,2.0 支持光标定位。
4. 在绑定数据,做表的分页.Update,Delete,等操作都可以可视化操作，方便了初学者。
5. 在 ASP.NET 中增加了 40 多个新的控件，减少了工作量。

[.NET(C#)]
什么是WSE？目前最新的版本是多少？
WSE (Web Services Enhancements) ，WSE可使开发人员跨安全平台建设可升级的、安全的Web服务，目前最新版本3.0。

[.NET(C#)]
如何部署一个ASP.net页面：
VS 2005和VS 2003都有发布机制。2003可以发布然后再复制部署。
VS2005 基本上可以直接部署到对应位置。

[.NET(C#)]
不定项选择：
以下叙述正确的是：
A. 接口中可以有虚方法。
B. 一个类可以实现多个接口。
C. 接口不能被实例化。
D. 接口中可以包含已实现的方法。
答案：B，C。

接口概述，接口具有下列属性：
接口类似于抽象基类：继承接口的任何非抽象类型都必须实现接口的所有成员。
不能直接实例化接口。
接口可以包含事件、索引器、方法和属性。
接口不包含方法的实现。
类和结构可从多个接口继承。
接口自身可从多个接口继承。

[.NET(C#)]
从数据库读取记录，你可能用到的方法有：
A. ExecuteNonQuery
B. ExecuteScalar
C. Fill
D. ExecuteReader
答案：B，C，D

[.NET(C#)]
如何把一个 Array 复制到 ArrayList 里？
(1) 实现1
```
string[] s ={ "111", "22222" };
ArrayList list = new ArrayList();
list.AddRange(s);
```

(2)实现2
```
string[] s ={ "111", "22222" };
ArrayList list = new ArrayList(s);
```

[.NET(C#)]
强名称：
强名称是由程序集的标识加上公钥和数字签名组成的，其中，程序集的标识包括简单文本名称、版本号和区域性信息（如果提供的话）。它使用对应的私钥从程序集文件中生成。（程序集文件包含程序集清单，其中包含组成程序集的所有文件的名称和哈希。）

[.NET(C#)]
C#中几种循环的方法，并指出他们的不同：
1. for：使用于确定次数的循环。
2. foreach：使用于遍历的元素是只读的。
3. while：次数不确定，条件随机变化。
4. do…while：次数不确定，条件随机变化，但至少要保证能被执行一次。

[.NET(C#)]
下列选项中，（）是引用类型？
a) enum类型
b) struct类型
c) string类型
d) int 类型
答案：C。

[.NET(C#)]
关于ASP.NET中的代码隐藏文件的描述正确的是（） ？
A. Web窗体页的程序的逻辑由代码组成，这些代码的创建用于与窗体交互。编程逻辑唯一与用户界面不同的文件中。该文件称作为“代码隐藏”文件，如果用C＃创建，该文件将具有“.ascx.cs”扩展名。
B. 项目中所有Web窗体页的代码隐藏文件都被编译成.EXE文件。
C. 项目中所有的Web窗体页的代码隐藏文件都被编译成项目动态链接库（.dll）文件。
D. 以上都不正确。
答案：C。

[.NET(C#)]
C#的数据类型有（B ）
A. 值类型和调用类型。
B. 值类型和引用类型。
C. 引用类型和关系类型。
D. 关系类型和调用类型。

[.NET(C#)]
下列描述错误的是（D ）
A. 类不可以多重继承而接口可以；
B. 抽象类自身可以定义成员而接口不可以；
C. 抽象类和接口都不能被实例化；
D. 一个类可以有多个基类和多个基接口；

[.NET(C#)]
在DOM中，装载一个XML文档的方法（B）
A. save方法
B. load方法
C. loadXML方法
D. send 方法

[.NET(C#)]
下列关于构造函数的描述正确的是（C）
A. 构造函数可以声明返回类型。
B. 构造函数不可以用private修饰
C. 构造函数必须与类名相同
D. 构造函数不能带参数

[.NET(C#)]
接口是一种引用类型，在接口中可以声明（ A ），但不可以声明公有的域或私有的成员变量。
A. 方法、属性、索引器和事件；
B. 方法、属性信息、属性；
C. 索引器和字段；
D. 事件和字段；

[.NET(C#)]
ASP.NET框架中，服务器控件是为配合Web表单工作而专门设计的。服务器控件有两种类型，它们是 （ A ）
A. HTML控件和Web控件
B. HTML控件和XML控件
C. XML控件和Web控件
D. HTML 控件和 IIS 控件

[.NET(C#)]
ASP.NET中，在Web窗体页上注册一个用户控件，指定该控件的名称为”Mike”，正确的注册指令为 （ B ）
A. <%@Register TagPrefix = “Mike” TagName = “Space2” Src = “myX.ascx”%>
B. <%@Register TagPrefix = “Space2” TagName = “Mike” Src = “myX.ascx”%>
C. <%@Register TagPrefix = “SpaceX” TagName = “Space2” Src = “Mike”%>
D. 以上皆非

[.NET(C#)]
在ADO.NET中，对于Command对象的ExecuteNonQuery()方法和ExecuteReader()方法，下面叙述错误的是（C ）。
A. insert、update、delete等操作的Sql语句主要用ExecuteNonQuery()方法来执行；
B. ExecuteNonQuery()方法返回执行Sql语句所影响的行数。
C. Select操作的Sql语句只能由ExecuteReader()方法来执行；
D. ExecuteReader()方法返回一个 DataReder 对象；

[.NET(C#)]
下列ASP.NET语句（B）正确地创建了一个与SQL Server 2000数据库的连接。
A. SqlConnection con1 = new Connection(“Data Source = localhost; Integrated Security = SSPI; Initial Catalog = myDB”);
B. SqlConnection con1 = new SqlConnection(“Data Source = localhost; Integrated Security = SSPI; Initial Catalog = myDB”);
C. SqlConnection con1 = new SqlConnection(Data Source = localhost; Integrated Security = SSPI; Initial Catalog = myDB);
D. SqlConnection con1 = new OleDbConnection(“Data Source = localhost; Integrated Security = SSPI; Initial Catalog = myDB”);

[.NET(C#)]
Winform中，关于ToolBar控件的属性和事件的描述不正确的是(D)。
A. Buttons属性表示ToolBar控件的所有工具栏按钮
B. ButtonSize属性表示ToolBar控件上的工具栏按钮的大小，如高度和宽度
C. DropDownArrows属性表明工具栏按钮（该按钮有一列值需要以下拉方式显示）旁边是否显示下箭头键
D. ButtonClick 事件在用户单击工具栏任何地方时都会触发

[.NET(C#)]
在ADO.NET中执行一个存储过程时，如果要设置输出参数则必须同时设置参数的方向和(D)，必要时还要设置参数尺寸。
A. 大小；
B. 上限；
C. 初始值；
D. 类型；

[.NET(C#)]
如果将窗体的FormBoderStyle设置为None，则(A)。
A. 窗体没有边框并不能调整大小；
B. 窗体没有边框但能调整大小；
C. 窗体有边框但不能调整大小；
D. 窗体是透明的；

[.NET(C#)]
如果要将窗体设置为透明的，则(B)
A. 要将FormBoderStyle属性设置为None;
B. 要将Opacity属性设置为小于100%得值;
C. 要将locked 属性设置为True;
D. 要将 Enabled 属性设置为 True;

[.NET(C#)]
下列关于C#中索引器理解正确的是(C)
A. 索引器的参数必须是两个或两个以上
B. 索引器的参数类型必须是整数型
C. 索引器没有名字
D. 以上皆非

[.NET(C#)]
下面描述错误的是(C)。
A. 窗体也是控件;
B. 窗体也是类;
C. 控件是从窗体继承来的;
D. 窗体的父类是控件类;

[.NET(C#)]
要对注册表进行操作则必须包含(D)。
A. System.ComponentModel命名空间;
B. System.Collections命名空间;
C. System.Threading命名空间;
D. Microsoft.Win32 命名空间;

[.NET(C#)]
要创建多文档应用程序，需要将窗体的(D)属性设为true。
A. DrawGrid;
B. ShowInTaskbar;
C. Enabled;
D. IsMdiContainer;

[.NET(C#)]
如果设treeView1=new TreeView()，则treeView1.Nodes.Add("根节点")返回的是一个 (A) 类型的值。
A. TreeNode;
B. int;
C. string;
D. TreeView;

[.NET(C#)]
下面关于XML的描述错误的是（ D ）。
A. XML提供一种描述结构化数据的方法；
B. XML 是一种简单、与平台无关并被广泛采用的标准；
C. XML文档可承载各种信息；
D. XML 只是为了生成结构化文档；

[.NET(C#)]
以下的C#代码，试图用来定义一个接口：

```
public interface IFile
{
    int A;
    int delFile()
    {
        A = 3;
    }
    void disFile();
}
```

关于以上的代码，以下描述错误的是(D)。
A. 以上的代码中存在的错误包括：不能在接口中定义变量，所以int A代码行将出现错误；
B. 以上的代码中存在的错误包括：接口方法delFile是不允许实现的，所以不能编写具体的实现函数；
C. 代码void disFile();声明无错误，接口可以没有返回值；
D. 代码 void disFile();应该编写为 void disFile(){}；

[.NET(C#)]
在ASP.NET中有Button控件myButton,要是单击控件时,导航到其他页面http://www.abc.com,正确的代码为(C)。
A. private void myButton_Click(object sender,
System.EventArgs e){Redirect(“http://www.abc.com”);}
B. private void myButton_Click(object sender,
System.EventArgs e){Request.Redirect(“http://www.abc.com”);}
C. private void myButton_Click(object sender,
System.EventArgs e){Reponse.Redirect(“http://www.abc.com”);}
D. private void myButton_Click(object sender,
System.EventArgs e){Request.Redirect(“http://www.abc.com”);return true;}

[.NET(C#)]
XML文档既包含数据同时也可包含（ B ）。
A. 元数据；
B. 架构；
C. 代码；
D. 图片；

[.NET(C#)]
在C#中利用Socket进行网络通信编程的一般步骤是：建立Socket侦听、（ A ）、利用Socket接收和发送数据。
A. 建立Socket连接；
B. 获得端口号；
C. 获得IP地址；
D. 获得主机名；

[.NET(C#)]
声明一个委托public delegate int myCallBack(int x); 则用该委托产生的回调方法的原型应该是（ B ）。
A. void myCallBack(int x) ；
B. int receive(int num) ；
C. string receive(int x) ；
D. 不确定的；

[.NET(C#)]
判断题，对的打√错的打×（每小题2分，共20分）
1. （T）在C#中，装箱操作是将值类型转化成引用类型。
2. （F）接口中的成员不可以有访问域修饰符，但可以有其它修饰符。
3. （F）在C#中，索引器是专门用来访问对象中的数组信息的。
4. （T）在C#中，接口可以被多重继承而类不能。
5. （F）在C#中，int[][]是定义一个int型的二维数组。
6. （T）异常类对象均为System. Exception类的对象。
7. （T）当窗体最小化后，再次还原成为活动窗体时将自动触发Paint事件。
8. （T）ASP.NET中，使用验证控件来验证用户输入，要求用户不可跳过该项输入，并且用户输入值在0和1000之间，则适用RequiredFieldValidator和RangeValidator控件。
9. （F）声明委托实际上是声明了一个方法。
10. （T）任何事物都是对象。

[.NET(C#)]
请简述一下用Socket进行同步通讯编程的详细步骤
1. 在应用程序和远程设备中使用协议和网络地址初始化套接字。
2. 在应用程序中通过指定端口和地址建立监听。
3. 远程设备发出连接请求。
4. 应用程序接受连接产生通信 scoket。
5. 应用程序和远程设备开始通讯（在通讯中应用程序将挂起直到通讯结束）。
6. 通讯结束，关闭应用程序和远程设备的 Socket 回收资源。

[.NET(C#)]
try {}里有一个return语句，那么紧跟在这个try后的finally {}里的code会不会被执行，什么时候被执行，在return前还是后?
答：会执行，在return前执行。

[.NET(C#)]
写出程序的输出结果：

```
public class A
{
    public virtual void Fun1( int i )
    {
        Console.WriteLine( i );
    }
    public void Fun2( A a )
    {
        a.Fun1( 1 );
        Fun1( 5 );
    }
}
public class B : A
{
    public override void Fun1( int i )
    {
        base.Fun1( i + 1 );
    }
    public static void Main()
    {
        B b = new B();
        A a = new A();
        a.Fun2( b );
        b.Fun2( a );
    }
}
```

答案：
2
5
1
6

[.NET(C#)]
分析以下代码，完成填空

```
string strTmp = "abcdefg某某某"; //中间无空格
int i = System.Text.Encoding.Default.GetBytes( strTmp ).Length;
int j = strTmp.Length;
```

以上代码执行完后，i= ？， j= ？
答：i=13,j=10
这个道题需要注意字符串中是否带空格，如果带空格，那么i、j的值应该增加，具体情况具体分析。

[.NET(C#)]
一列数的规则如下：1、1、2、3、5、8、13、21、34..... ；求第30位数是多少，用递归算法实现。(C#语言)
答案：

```
class Program
{
    public static void Main()
    {
        Console.WriteLine( Foo( 30 ) );
    }
    public static int Foo( int i )
    {
        if( i <= 0 )
            return 0;
        else if( i > 0 && i <= 2 )
            return 1;
        else
            return Foo( i - 1 ) + Foo( i - 2 );
    }
}
```

[.NET(C#)]
请编程遍历页面上所有TextBox控件并给它赋值为string.Empty
答案：

```
foreach( System.Windows.Forms.Control control in this.Controls )
{
    if( control is System.Windows.Forms.TextBox )
    {
        System.Windows.Forms.TextBox txtbox =
        ( System.Windows.Forms.TextBox )control;

        txtbox.Text = string.Empty;
    }
}
```

[.NET(C#)]
公司要求开发一个继承System.Windows.Forms.ListView类的组件，要求达到以下的特殊功能：点击ListView各列列 头时，能按照点击列的每行值进行重排视图中的所有行 (排序的方式如DataGrid相似)。根据您的知识，请简要谈一下您的思路：
答：根据点击的列头,包该列的 ID 取出,按照该 ID 排序后,在给绑定到 ListView 中。

[.NET(C#)]
String s = new String("xyz");创建了几个 String Object?
答：两个对象，一个是“xyx”，一个是指向“xyx”的引用对像 s。

[.NET(C#)]
请编程实现一个冒泡排序算法
答案：

```
int[] array = new int[10];
int temp = 0;

public void Po()
{
    for( int i = 0; i < array.Length - 1; i++ )
    {
        for( int j = i + 1; j < array.Length; j++ )
        {
            if( array[j] < array[i] )
            {
                temp = array[i];
                array[i] = array[j];
                array[j] = temp;
            }
        }
    }
}
```

[.NET(C#)]
求以下表达式的值，写出您想到的一种或几种实现方法：1 - 2 + 3 - 4 +.....+ m
答案：

```
public static void Main()
{
    int input = int.Parse( Console.ReadLine() );
    int sum = 0;
    for( int i = 0; i <= input; i++ )
    {
        if( ( i % 2 ) == 1 )
        {
            sum += i;
        }
        else
        {
            sum = sum - i;
        }
    }
    Console.WriteLine( sum );
}
```

或：

```
int m = int.Parse( Console.ReadLine() );
int sum = 0;
bool flag = true;
for( int i = 1; i <= m; i++ )
{
    if( flag )
        sum += i;
    else
        sum -= i;
    flag = !flag;
}
Console.WriteLine( sum );
```

[.NET(C#)]

在下面的例子里

```
    class A
    {
        public A()
        {
            PrintFields();
        }
        public virtual void PrintFields()
        {
        }
    }

    class B : A
    {
        int x = 1;
        int y;
        public B()
        {
            y = -1;
        }
        public override void PrintFields()
        {
            Console.WriteLine( "x={0},y={1}", x, y );
        }
    }
```

当使用new B()创建B的实例时，产生什么输出？

答案：x = 1, y = 0

[.NET(C#)]
根据委托(delegate)的知识，请完成以下用户控件中代码片段的填写：

```
namespace test
{
    public delegate void OnDBOperate();
    public class UserControlBase : System.Windows.Forms.UserControl
    {
        public event OnDBOperate OnNew;
        private void toolBar_ButtonClick( object sender,
            System.Windows.Forms.ToolBarButtonClickEventArgs e )
        {
            if( e.Button.Equals( BtnNew ) )
            {
                //请在以下补齐代码用来调用OnDBOperate委托签名的OnNew事件。
            }
        }
    }
}
```

答案：

```
    if( OnNew != null )
        OnNew();
```

注意：此处注意审题，`public delegate void OnDBOperate();`是没有参数的，所以要写`OnNew()`。网上有的答案是：`OnNew( this, e );`。这个是错误的，一定要注意`OnDBOperate()`是否带参数。

[.NET(C#)]
根据线程安全的相关知识，分析以下代码，当调用 test 方法时 i>10 时是否会引起死锁?并简要说明理由。

```
public void test( int i )
{
    lock( this )
    {
        if( i > 10 )
        {
            i--;
            test( i );
        }
    }
}
```

答案：不会死锁。

[.NET(C#)]
两个对象值相同(x.equals(y) == true)，但却可有不同的hash code，这句话对不对?
答：不对，有相同的hash code。

[.NET(C#)]
给定以下 XML 文件，完成算法流程图

```
<FileSystem>
  <DriverC>
    <Dir DirName="MSDOS622">
      <File FileName =" Command.com" ></File>
    </Dir>
    <File FileName ="MSDOS.SYS" ></File>
    <File FileName =" IO.SYS" ></File>
  </DriverC>
</FileSystem>
```

请画出遍历所有文件名（FileName）的流程图(请使用递归算法)。

```
public void DomDepthFirst( XmlNode currentNode )
{
    XmlNode node = currentNode.FirstChild;
    while( node != null )
    {
        if( node.Name == "File" )
        {
            Console.Write( ( ( XmlElement )node ).
                GetAttribute( "FileName" ) + "/r/n" );
        }
        DomDepthFirst( node );
        node = node.NextSibling;
    }
}
```

[.NET(C#)]
swtich 是否能作用在 byte 上，是否能作用在 long 上，是否能作用在 String 上?
答案：可以作用在 byte 和 long 上，也可以作用在 string 上。

[.NET(C#)]
当一个线程进入一个对象的一个 synchronized 方法后，其它线程是否可进入此对象的其它方法?
答案：不能，一个对象的一个 synchronized 方法只能由一个线程访问。

[.NET(C#)]
short s1 = 1; s1 = s1 + 1;有什么错? short s1 = 1; s1 += 1;有什么错?
答：

1. short s1 = 1; s1 = s1 + 1;有错，s1 是 short 型，s1+1 是 int 型，不能隐式转化为 short 型。可修改为 s1 =(short)(s1 + 1) 。
2. short s1 = 1; s1 += 1 正确。

[.NET(C#)]
问题 1：产生一个 int 数组，长度为 100，并向其中随机插入 1-100，并且不能重复。
答案：

```
int[] intArr = new int[100];
ArrayList myList = new ArrayList();
Random rnd = new Random();

while( myList.Count < 100 )
{
    int num = rnd.Next( 1, 101 );
    if( !myList.Contains( num ) )
    {
        myList.Add( num );
    }
}
for( int i = 0; i < 100; i++ )
{
    intArr[i] = ( int )myList[i];
}
```

问题 2：对上面生成的数组排序，需要支持升序、降序两种顺序。
答案：ArrayList.Sort()方法可以实现排序。

[.NET(C#)]
需要实现对一个字符串的处理，首先将该字符串首尾的空格去掉，如果字符串中间还有连续空格的话，仅保留一个空格，即允许字符串中间有多个空格，但连续的空格数不可超过一个。
答案：

```
string inputStr = " xx xx ";
inputStr = Regex.Replace( inputStr.Trim(), @"/s+", " " );
```

[.NET(C#)]
下面这段代码输出什么？为什么？

```
int i = 5;
int j = 5;
if( Object.ReferenceEquals( i, j ) )
    Console.WriteLine( "Equal" );
else
    Console.WriteLine( "Not Equal" );
```

答案：Not Equal。因为比较的是对象

[.NET(C#)]
public static const int A=1;这段代码有错误么？是什么？
答：const 不能用 static 修饰。

[.NET(C#)]
float f = -123.567F; int i = (int)f; i 的值现在是_____?
答：-123。

[.NET(C#)]
下面的代码中有什么错误吗?

```
class A
{
    public virtual void F()
    {
        Console.WriteLine( "A.F" );
    }
}
abstract class B : A
{
    public abstract override void F();//abstract与override关键字可以同时使用
}
```

答案：没有错误！可以通过编译器。
注意：网上有资料说 abstract 与 override 关键字不可以同时使用，这种说法是错误的！

[.NET(C#)]
下面这段代码有错误么?

```
switch( i )
{
    case ():        //答：case()条件不能为空。
        CaseZero();
        break;
    case 1:
        CaseOne();
        break;
    case 2:
        dufault;    //答：格式不正确。正确写法 default:
        CaseTwo();
        break;
}
```

[.NET(C#)]
在.net（C# or vb.net）中如何取消一个窗体的关闭。
答案：在VS2005下

```
private void Form1_FormClosing( object sender, FormClosingEventArgs e )
{
    e.Cancel = false;
}
```

注意：有的写答案写“private void Form1_Closing”事件，如果在VS2005下，这个答案是错误的，VS2005没有这个事件。

[.NET(C#)]
在.net（C# or vb.net）中如何用户自定义消息，并在窗体中处理这些消息。
答案：

```
private const int WM_Lbutton = 0x001C;
protected override void WndProc( ref Message m )
{
    switch( m.Msg )
    {
        case WM_Lbutton:
        string message = string.Format( "收到消息!参数为:{0},{1}",
                                        m.WParam, m.LParam );
        MessageBox.Show( message );
        break;
    }
    base.WndProc( ref m );
}
```

[.NET(C#)]
在 C#中有一个 double 型的变量，比如 10321.5，比如 122235401.21644，作为货币的值如何按各个不同国家的习惯来输出。比如美国用$10,321.50 和$122，235，401.22 而在英国则为 £10 321.50 和 £122 235 401.22
答案:

```
System.Globalization.CultureInfo MyCulture = new
System.Globalization.CultureInfo( "en-US" );//美国

System.Globalization.CultureInfo MyCulture = new
System.Globalization.CultureInfo( "en-GB" ); //英国

decimal y = 99999999999999999999999999m;
string str = String.Format( MyCulture, "My amount = {0:c}", y );
```

[.NET(C#)]

写一个 HTML 页面，实现以下功能，左键点击页面时显示“您好”，右键点击时显示“禁止右键”。并在 1 秒后自动关闭页面。

```
<html>
<head>
    <script language="javascript">
        function MyClick()
        {
            if( event.button  ==  1 )
            {
                 alert( "您好");
            }
            else if(event.button  ==  2 )
            {
                 alert( "禁止右键");
                 setTimeout( "MyClose(); ",    1000);
            }
        }
        function MyClose()
        {
            window.close();
        }
    </script>
</head>
<body onmouseup="MyClick();">
   abc
</body>
</html>
```

[.NET(C#)]

分析以下代码。

```
public static void test( string ConnectString )
{
    System.Data.OleDb.OleDbConnection conn = new
System.Data.OleDb.OleDbConnection();
    conn.ConnectionString = ConnectString;
    try
    {
        conn.Open();
        //„„.
    }
    catch( Exception Ex )
    {
        MessageBox.Show( Ex.ToString() );
    }
    finally
    {
        if( !conn.State.Equals( ConnectionState.Closed ) )
            conn.Close();
    }
}
```

问题：

1. 以上代码可以正确使用连接池吗？
   答：如果传入的 connectionString 是正确的格式的话，可以正确使用连接池。
2. 以上代码所使用的异常处理方法，是否所有在 test 方法内的异常都可以被捕捉并显示出来？
   答：不能捕获所有的异常，只能捕获在 try 块内发生的异常。

[.NET(C#)]

以下是一些C#中的枚举型的定义，其中错误的用法有（B）

A. public enum var1{ Mike = 100, Nike = 102, Jike }
B. public enum var1{ Mike = “1”, Nike, Jike }
C. public enum var1{ Mike=-1 , Nike, Jike }
D. public enum var1{ Mike , Nike , Jike }

[.NET(C#)]
下面的例子中

```
class A
{
    public static int X;
    static A()
    {
        X = B.Y + 1;
    }
}
class B
{
    public static int Y = A.X + 1;
    static B()
    {
    }
    static void Main()
    {
        Console.WriteLine( "X={0},Y={1}", A.X, B.Y );
    }
}
```

产生的输出结果是什么？
答：x=1,y=2

[.NET(C#)]
写出程序的输出结果

```
class Class1
{
    private string str = "Class1.str";
    private int i = 0;
    static void StringConvert( string str )
    {
        str = "string being converted.";
    }
    static void StringConvert( Class1 c )
    {
        c.str = "string being converted.";
    }
    static void Add( int i )
    {
        i++;
    }
    static void AddWithRef( ref int i )
    {
        i++;
    }
    static void Main()
    {
        int i1 = 10;
        int i2 = 20;
        string str = "str";
        Class1 c = new Class1();
        Add( i1 );
        AddWithRef( ref i2 );
        Add( c.i );
        StringConvert( str );
        StringConvert( c );
        Console.WriteLine( i1 );
        Console.WriteLine( i2 );
        Console.WriteLine( c.i );
        Console.WriteLine( str );
        Console.WriteLine( c.str );
    }
}
```

答案：10, 21, 0, str, string being converted
注意：此处加逗号“, ”是为了答案看起来清晰，实际结果是纵向排列的，因为调用了Console.WriteLine()。

[.NET(C#)]

写出程序的输出结果

```
public abstract class A
{
    public A()
    {
        Console.WriteLine( 'A' );
    }
    public virtual void Fun()
    {
        Console.WriteLine( "A.Fun()" );
    }
}
public class B : A
{
    public B()
    {
        Console.WriteLine( 'B' );
    }
    public new void Fun()
    {
        Console.WriteLine( "B.Fun()" );
    }
    public static void Main()
    {
        A a = new B();
        a.Fun();
    }
}
```

答案：

A

B

A.Fun()