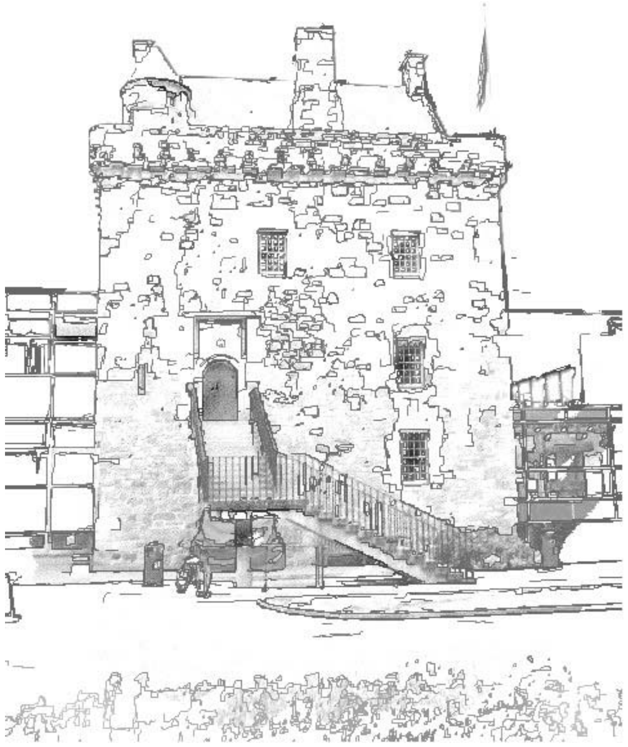
Introduction to .NET

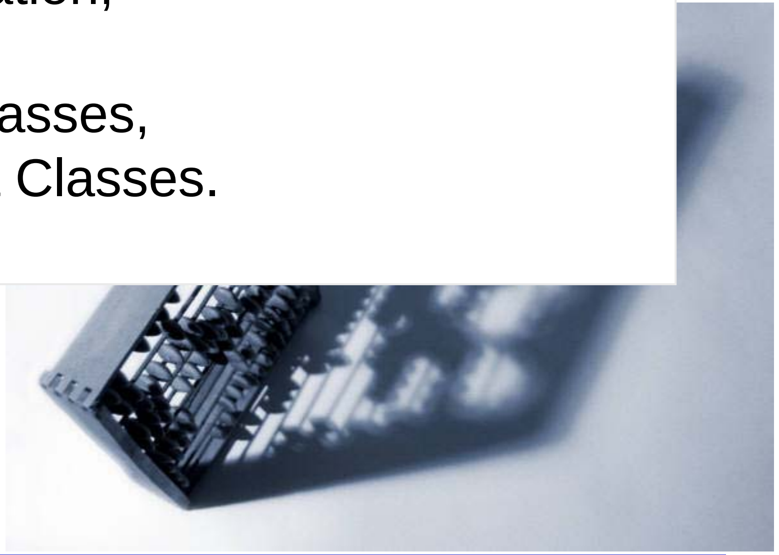


Andrew Cumming, SoC

Bill Buchanan, SoC

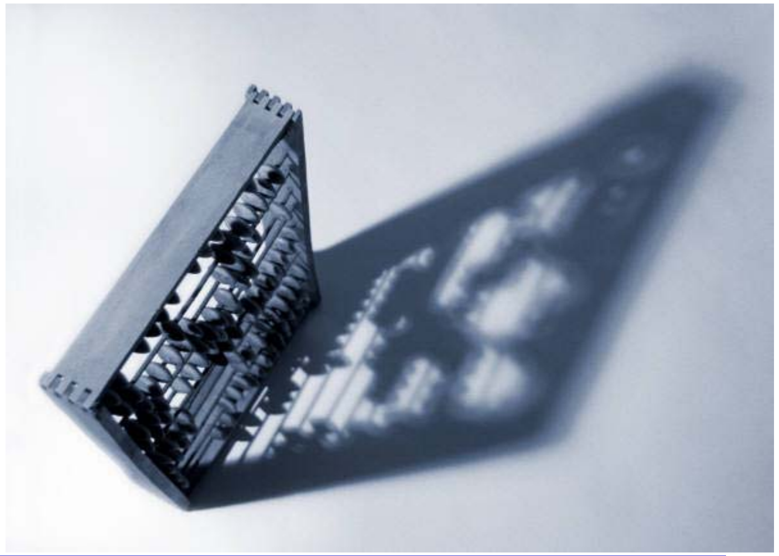
Course Outline

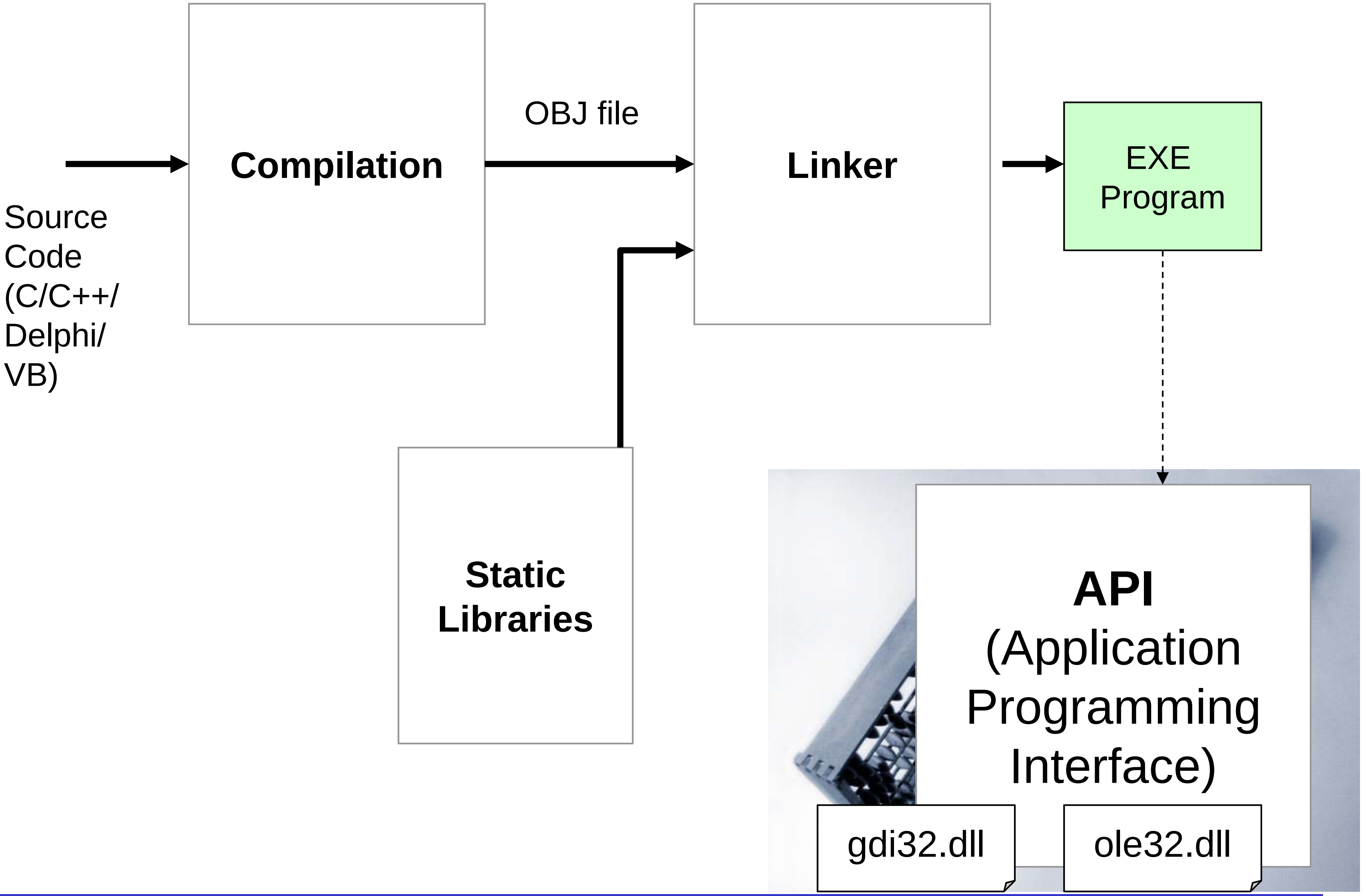
11-12am	Introduction to .NET, Overview of .NET Framework, .NET Components, C#.
12-1pm:	C# Language Elements
1-1:45pm	Classes, Encapsulation, Object-Orientation, Classes, Sealed Classes, Interfaces, Abstract Classes.
1:45-2pm:	Certification



Module 1

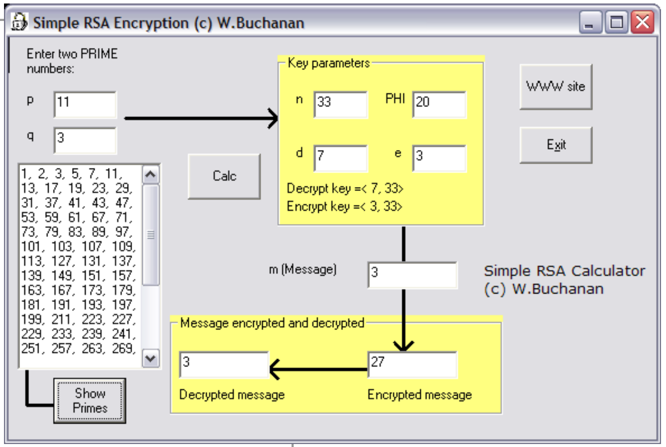
- .NET Framework.
- Visual Studio Environment.
- Benefits of C# over VB.
- .NET Components.
- .NET Languages.




Source Code (C/C++/ Delphi/ VB)
Compilation
OBJ file
Linker
EXE Program
Static Libraries
API (Application Programming Interface)
gdi32.dll
ole32.dll

Traditional Windows Model

Creating windows.
Windows support functions.
Message processing.
Menus.
Resources.
Dialog boxes.
User input functions.
Memory management.
GDI (graphical device interface).
Bitmaps, icons and metafiles.
Printing and text output.
Painting and drawing.
File I/O.
Clipboard. Support for public and private clipboards.
Registry. Support for functions which access the Registry.
Initialization files. Support for functions which access INI files.
System information.
String manipulation.
Timers.
Processes and thre
Error and exception
MDI (multiple docu
Help files.
File compression/d
DLLs.
Network support (N
Multimedia support
OLE and DDE (dyr
TrueType fonts.


Simple RSA Encryption (c) W.Buchanan

Enter two PRIME numbers:
p: 11
q: 3

1, 2, 3, 5, 7, 11,
13, 17, 19, 23, 29,
31, 37, 41, 43, 47,
53, 59, 61, 67, 71,
73, 79, 83, 89, 97,
101, 103, 107, 109,
113, 127, 131, 137,
139, 149, 151, 157,
163, 167, 173, 179,
181, 191, 193, 197,
199, 211, 223, 227,
229, 233, 239, 241,
251, 257, 263, 269,

Key parameters
n: 33 PHI: 20
d: 7 e: 3
Decrypt key =< 7, 33>
Encrypt key =< 3, 33>

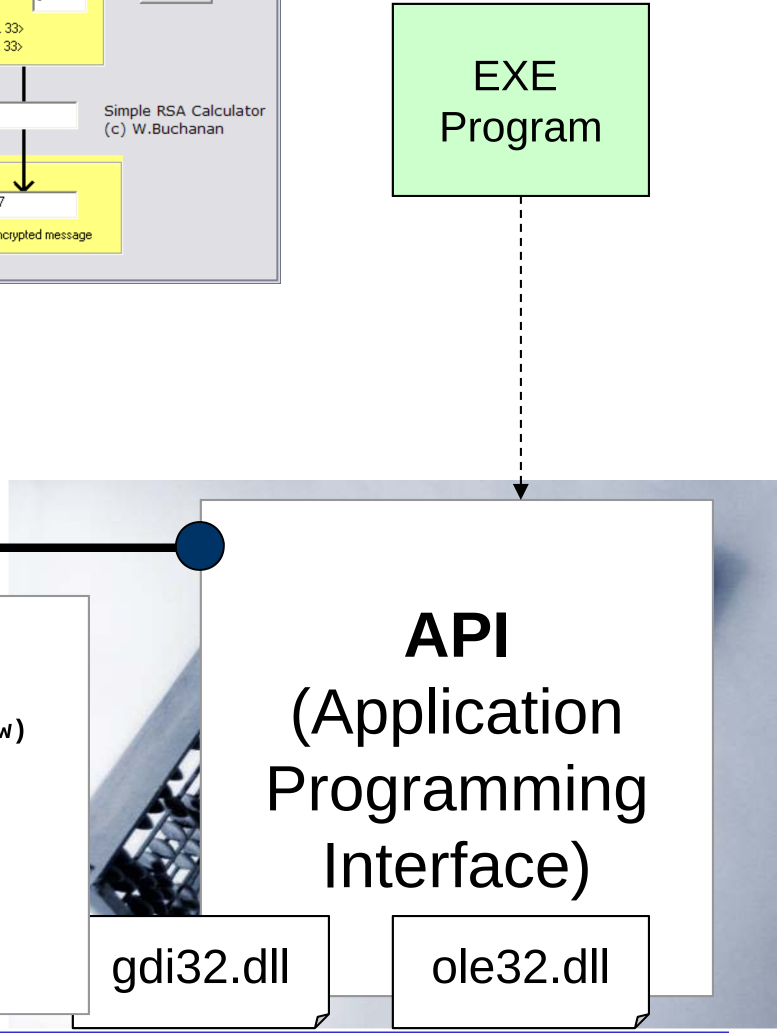
m (Message): 3

Message encrypted and decrypted
Decrypted message: 3
Encrypted message: 27

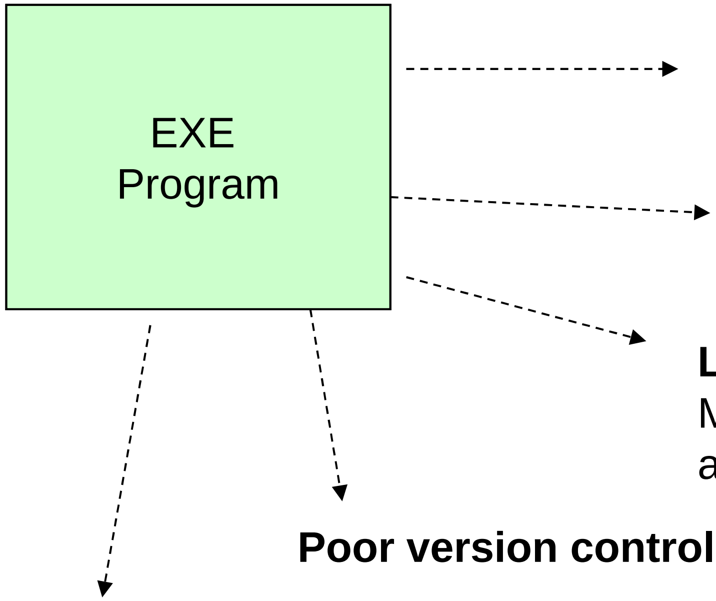
Simple RSA Calculator
(c) W.Buchanan

Example C++ code calling an API

```
#include <windows.h>
int WINAPI WinMain(HINSTANCE hInstance,
     HINSTANCE hPrev, LPSTR lpCmd,  int nShow)
{
char msg[128];
     wsprintf(msg, "My name is Fred");
     MessageBox(GetFocus(), msg, "My first
     Window", MB_OK | MB_ICONINFORMATION);
     return(0);
}
```



Traditional Windows Model



Lack of support for different hardware
Programs were compiled to x86 code.

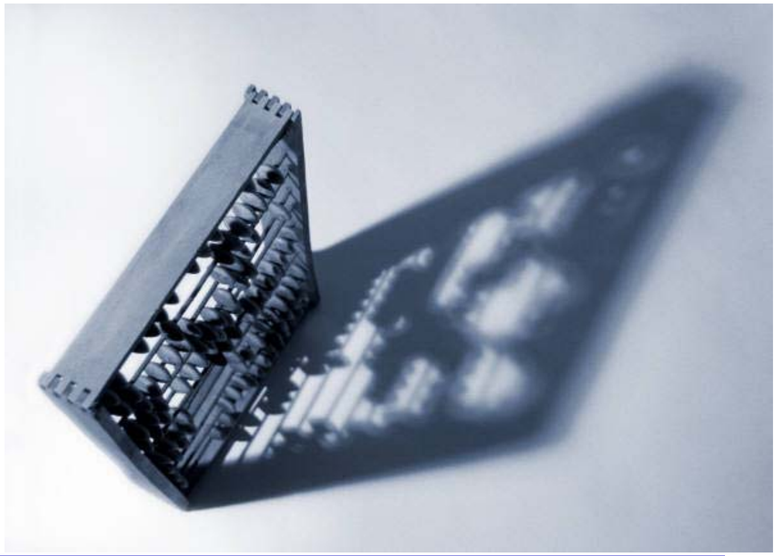
Difficult to integrate different languages

Lack of security Integration
Most programs where written with little care about security

Poor version control for system components

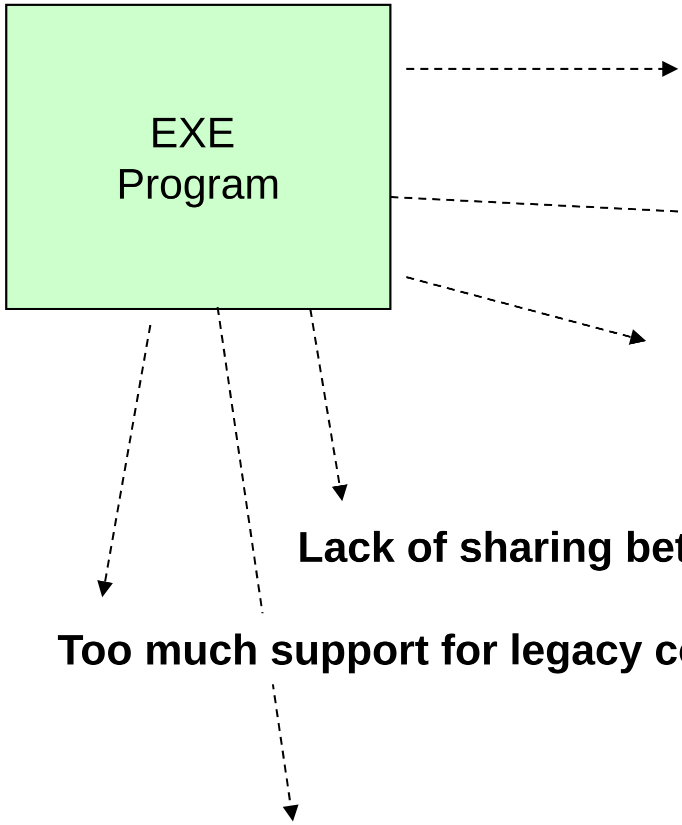
Weak integration with Internet/WWW
Code and WWW code where seen as separate entities.

```
Code.asp
<%
     val1 = 10
     val2 = 20
     result = Cstr(val1) + Cstr(val2)
     response.write "<BR>Value is " & result
     result = val1 + val2
     response.write "<BR>Value is " & result
%>
```



So what's wrong with the old model?

So what's wrong with the old model?



Poor robustness
Where applications often crash

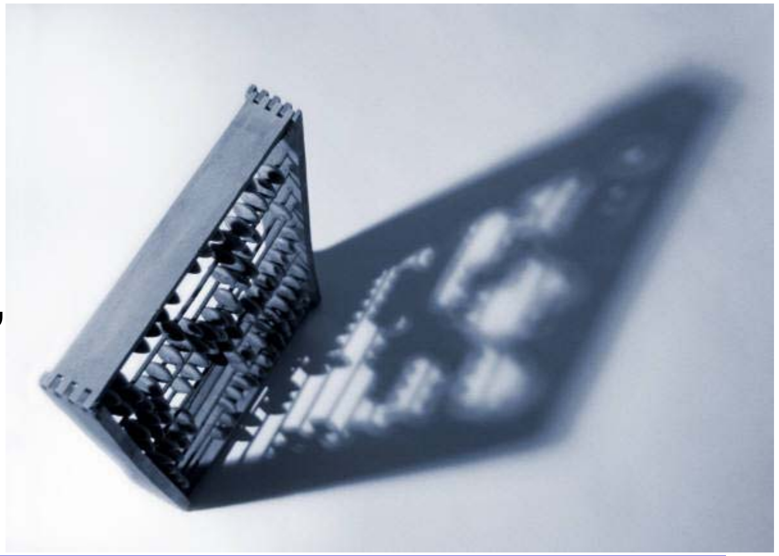
Difficult deployments

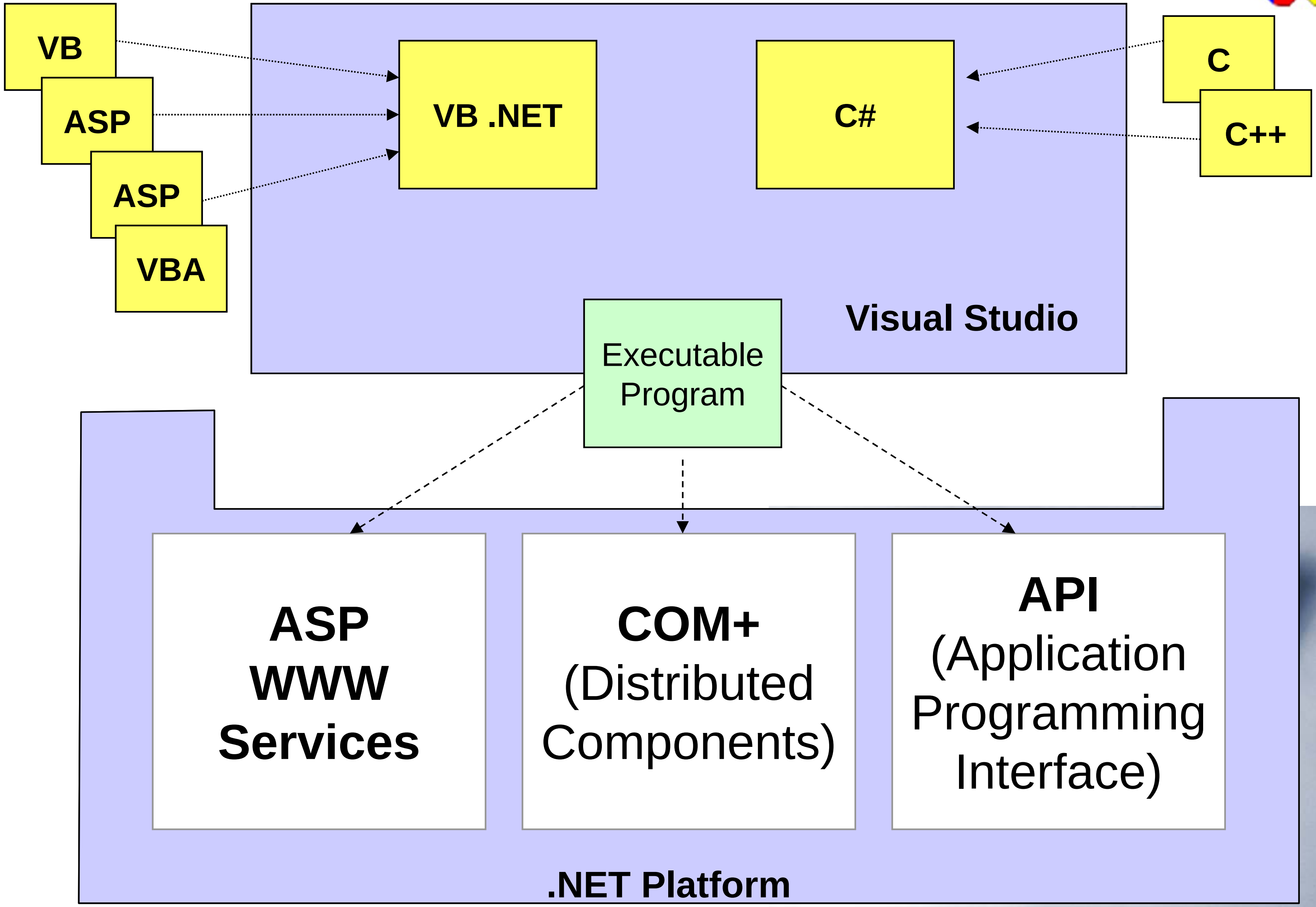
Poor pre-run checking
This is where applications often crashed on run-time

Lack of sharing between applications

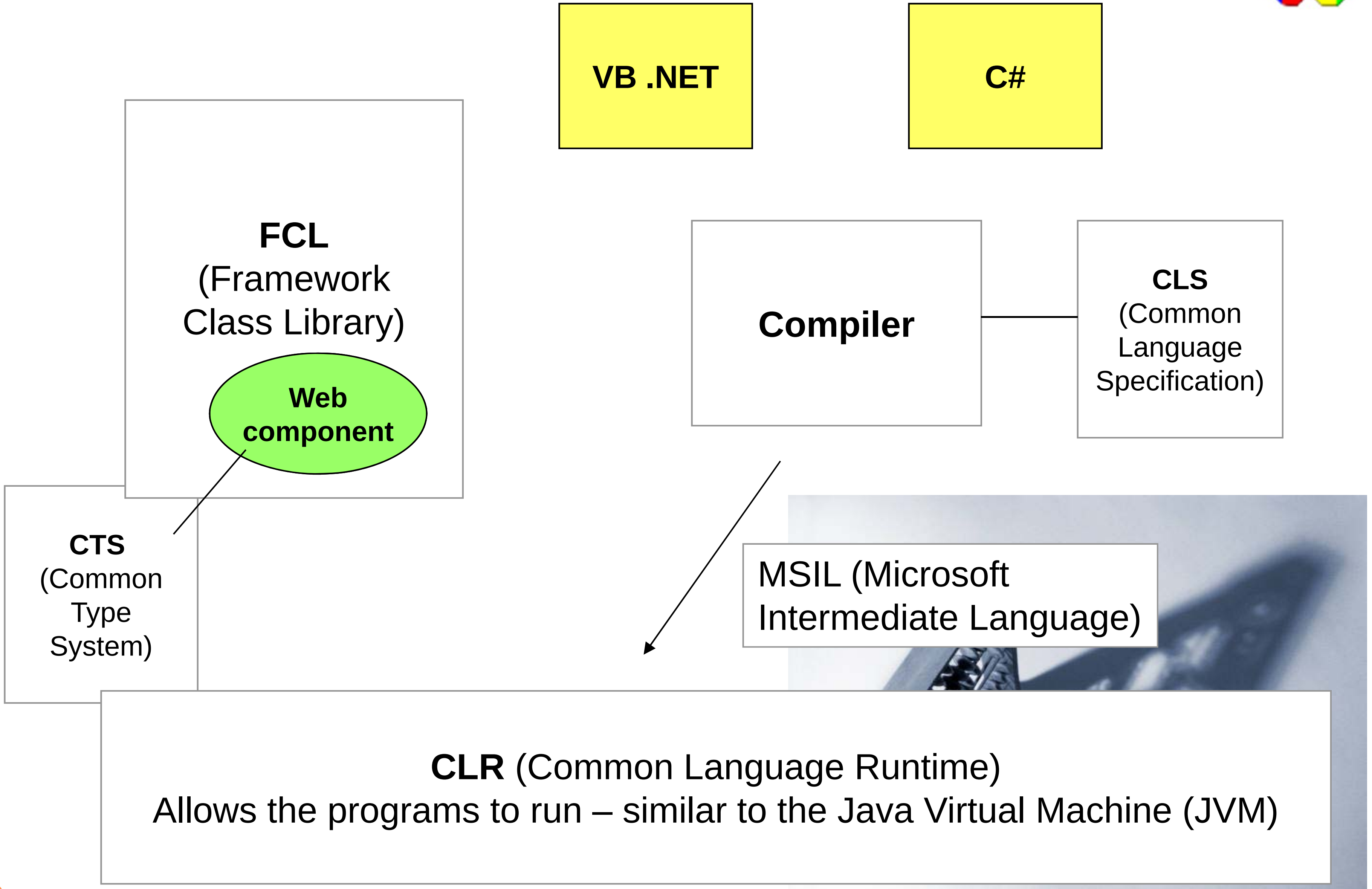
Too much support for legacy code

Poor integration with different data sources, such as XML, databases, and text files.





New .NET Platform



How are languages integrated?

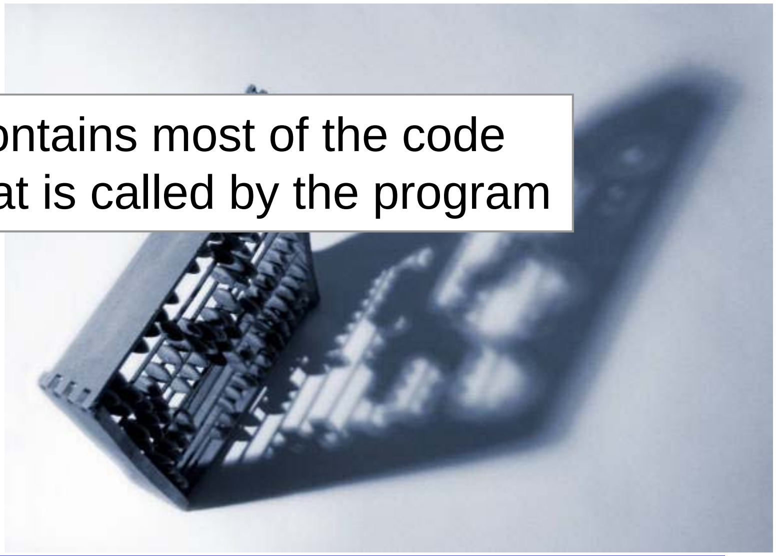
```
 Volume in drive C has no label.
 Volume Serial Number is 1A83-0D9D

 Directory of C:\WINDOWS\Microsoft.NET\Framework\v1.1.

21/02/2003  08:24              7,680 Accessibility.dll
21/02/2003  06:00             98,304 alink.dll
20/02/2003  20:19             24,576 aspnet_filter.dll
20/02/2003  20:19            253,952 aspnet_isapi.dll
20/02/2003  20:19             40,960 aspnet_rc.dll
20/02/2003  20:09             77,824 CORPerfMonExt.dll
21/02/2003  11:21            626,688 cscomp.dll
21/02/2003  08:24             12,288 cscompmgd.dll
21/02/2003  08:24             33,792 CustomMarshalers.dll
29/07/2002  12:11            219,136 c_g18030.dll
21/02/2003  11:21            524,288 diasymreader.dll
19/03/2003  02:52            245,760 envdte.dll
20/02/2003  20:16            798,720 EventLogMessages.dll
20/02/2003  20:06            282,624 fusion.dll
21/02/2003  08:24              7,168 IEExecRemote.dll
21/02/2003  08:24             32,768 IEHost.dll
21/02/2003  08:24              4,608 IIEHost.dll
21/02/2003  08:25          1,564,672 mscorcfg.dll
20/02/2003  20:09             77,824 mscordbc.dll
20/02/2003  20:09            233,472 mscordbi.dll
20/02/2003  20:09             86,016 mscorie.dll
20/02/2003  20:06            311,296 mscorjit.dll
20/02/2003  20:09             98,304 mscorld.dll
21/02/2003  08:26          2,088,960 mscorlib.dll
20/02/2003  19:43            131,072 mscormmc.dll
20/02/2003  20:06             65,536 mscorpe.dll
20/02/2003  20:09            143,360 mscorrc.dll
20/02/2003  20:09             81,920 mscorsec.dll
20/02/2003  20:09             77,824 mscorsn.dll
20/02/2003  20:07          2,494,464 mscorsvr.dll
```

Program

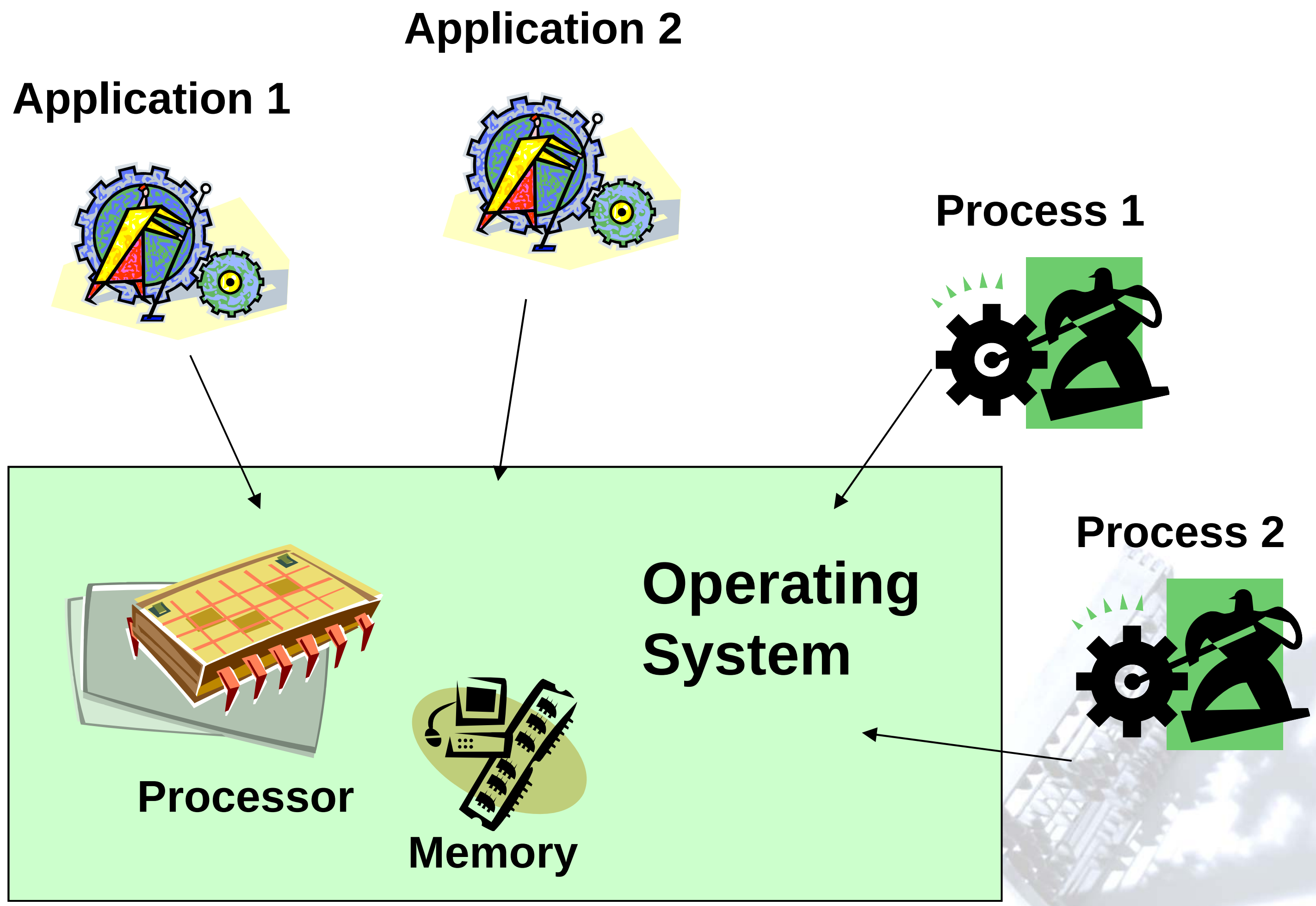
$\longrightarrow$

Mscorlib.dll
Arrays,
File I/O,
System,
Security

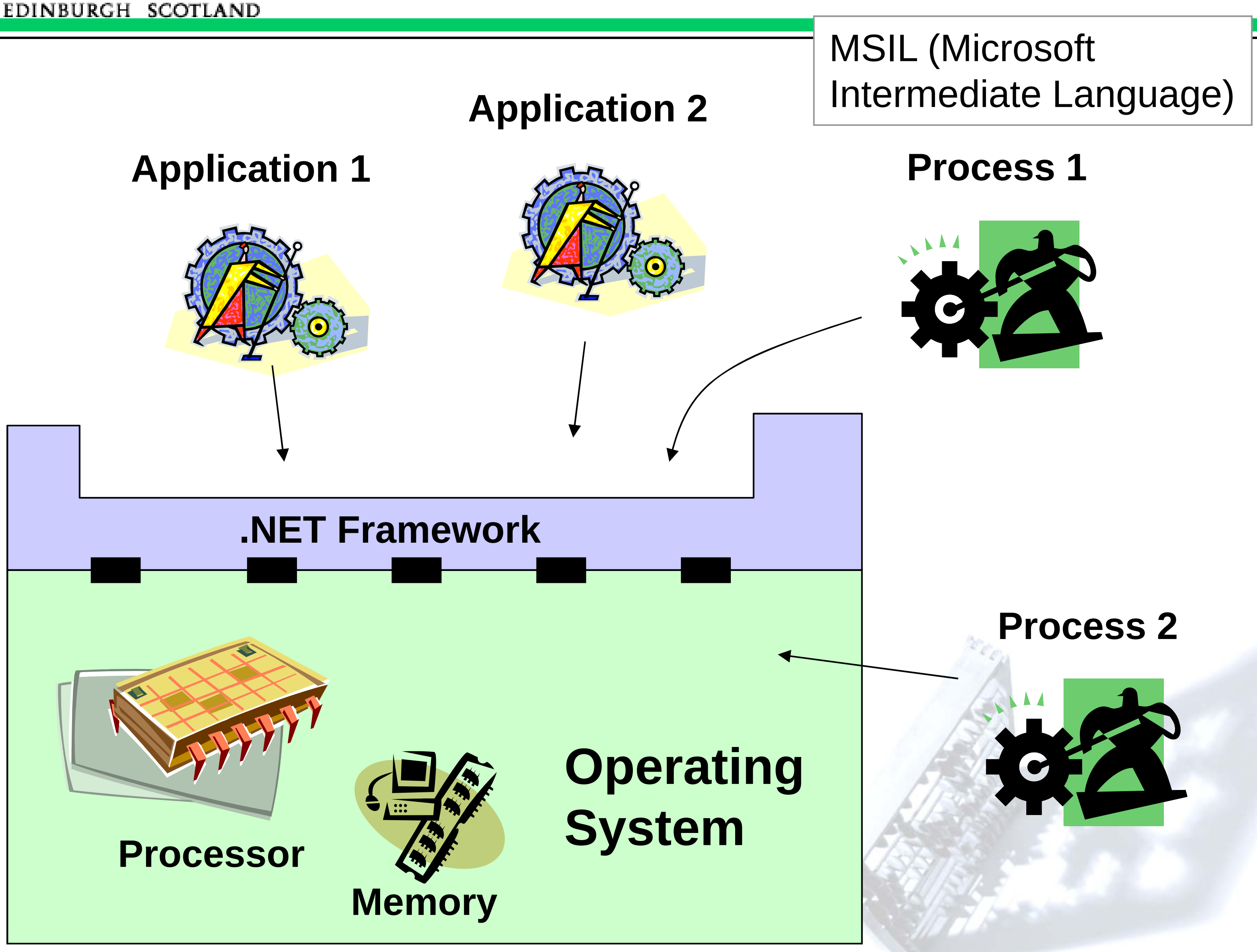
Contains most of the code that is called by the program



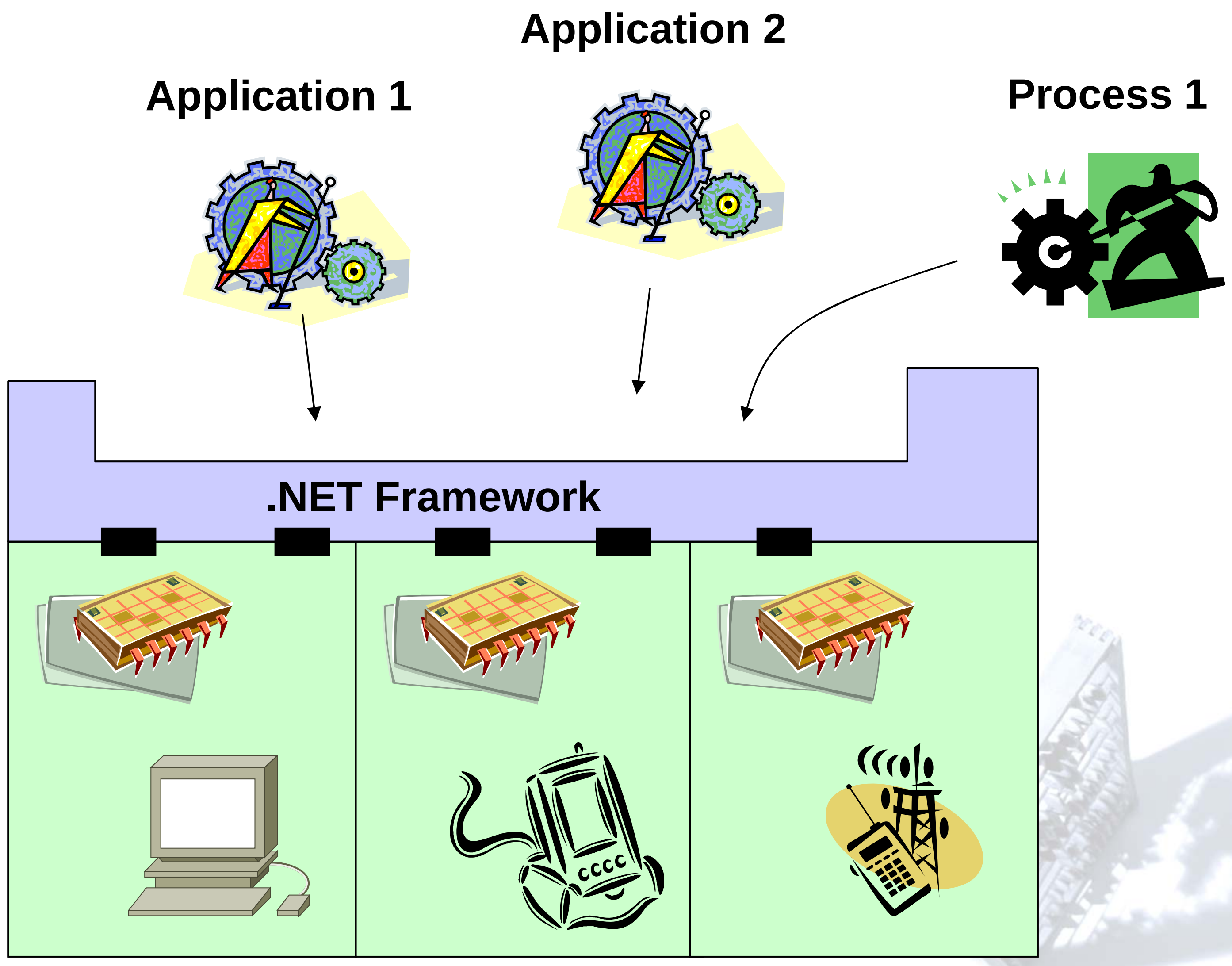
.NET Framework Files



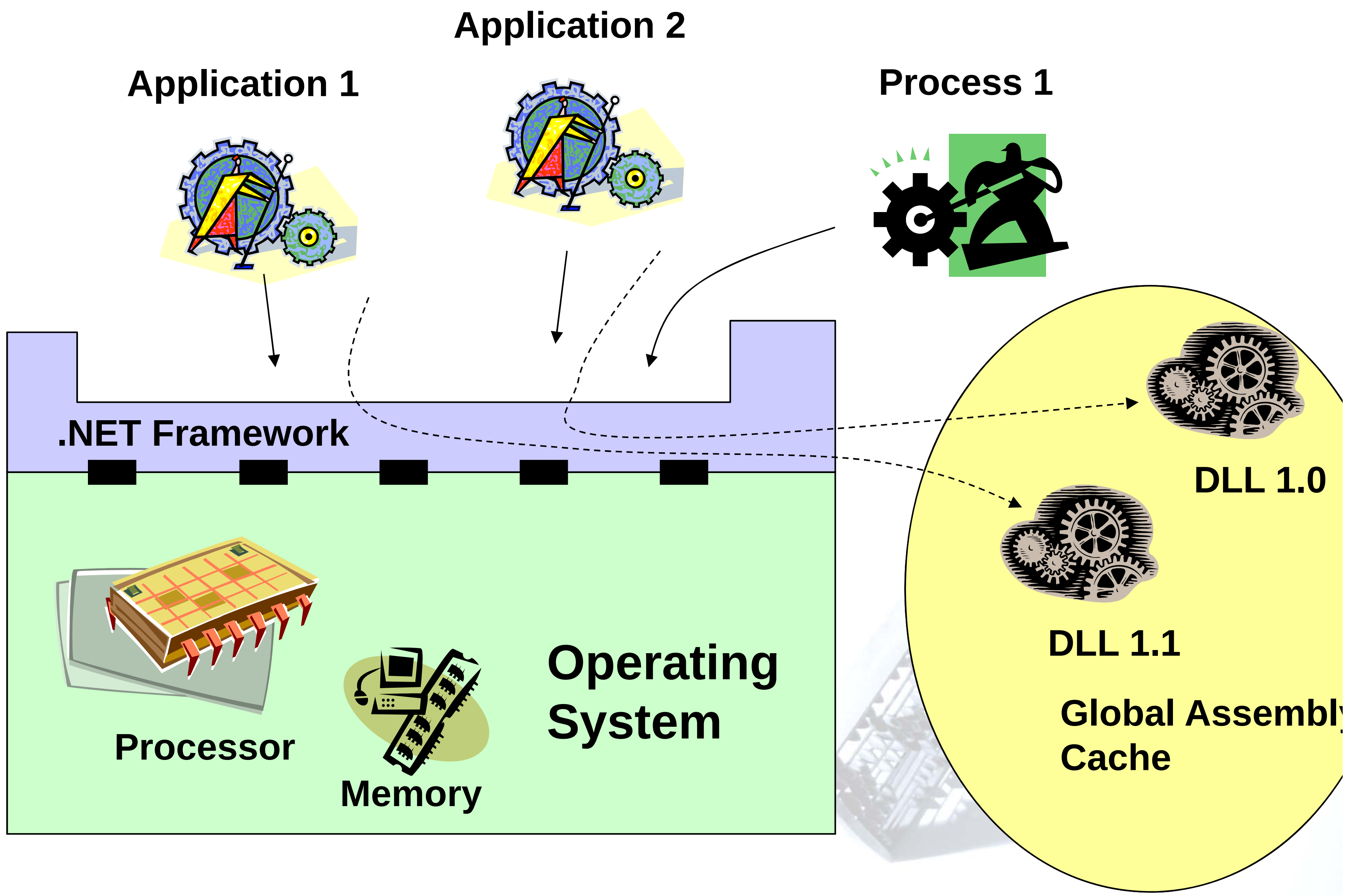
Traditional method of running applications and processes



.NET method of running applications and processes



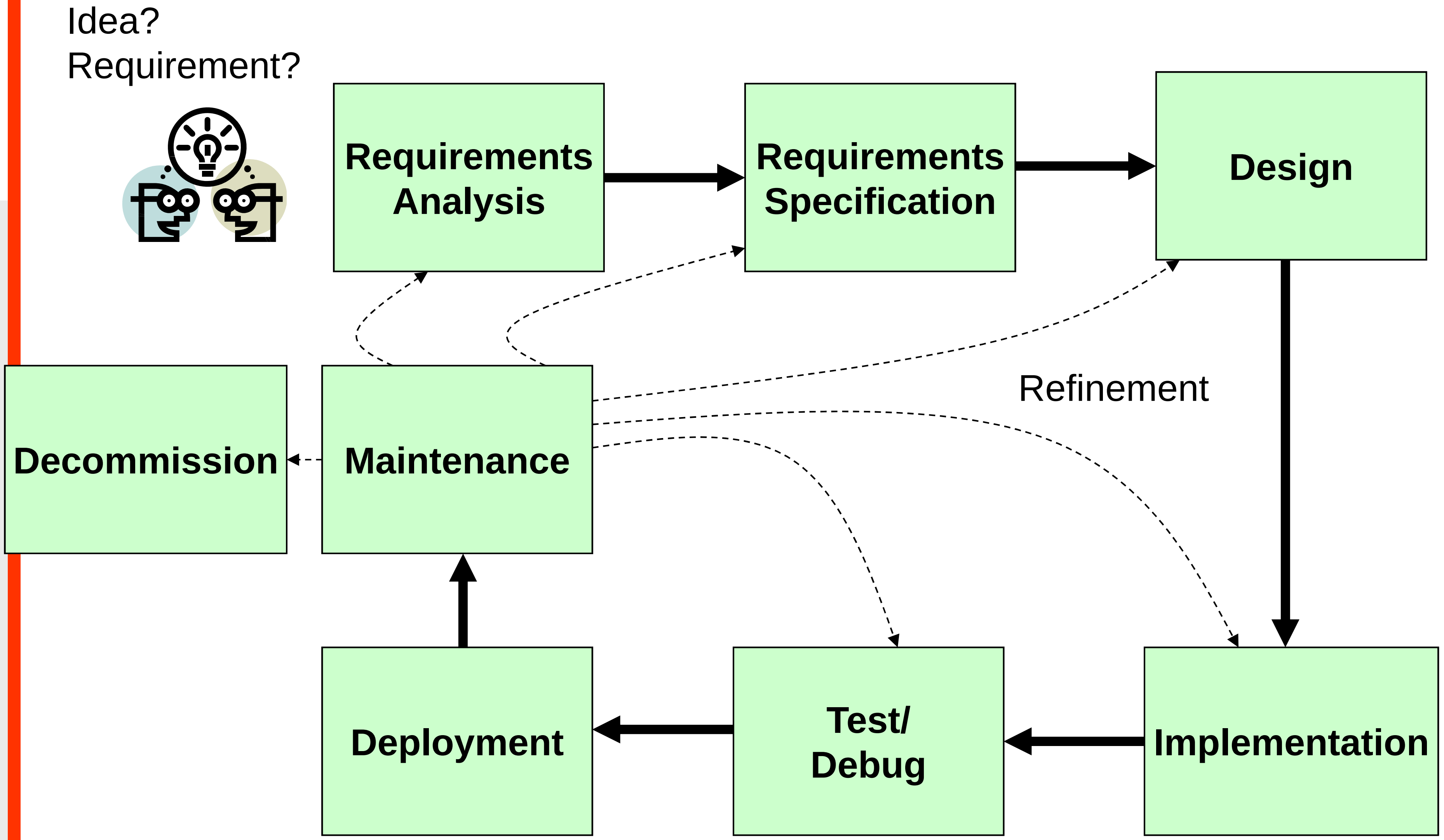
.NET method of running applications and processes



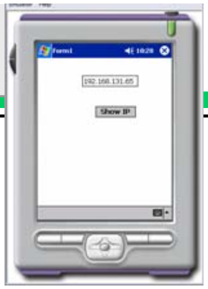
.NET method of running applications and processes

Visual Studio Environment

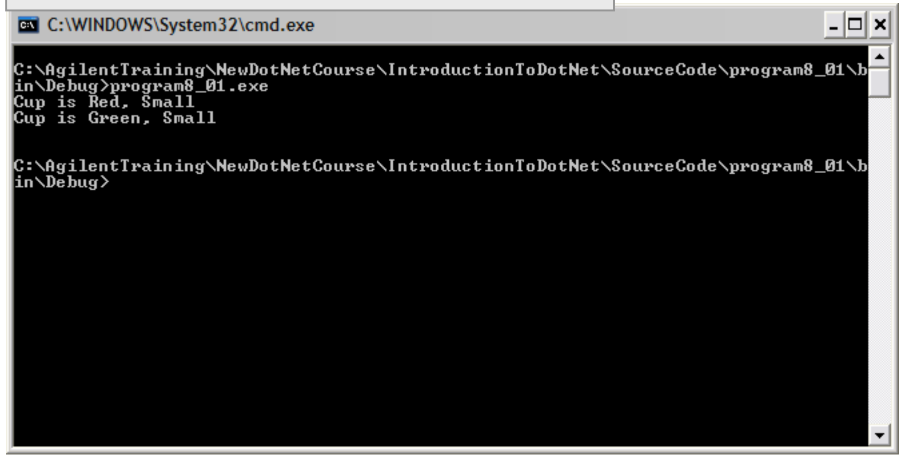
Bill Buchanan


Idea?
Requirement?
Requirements Analysis
Requirements Specification
Design
Refinement
Decommission
Maintenance
Deployment
Test/ Debug
Implementation

Mobile Application



Console Application



```
C:\AgilentTraining\NewDotNetCourse\IntroductionToDotNet\SourceCode\program8_01\b
in\Debug>program8_01.exe
Cup is Red, Small
Cup is Green, Small

C:\AgilentTraining\NewDotNetCourse\IntroductionToDotNet\SourceCode\program8_01\b
in\Debug>
```

Windows Service

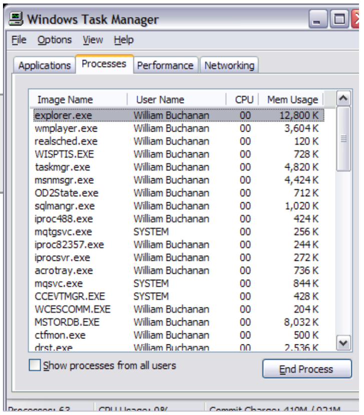
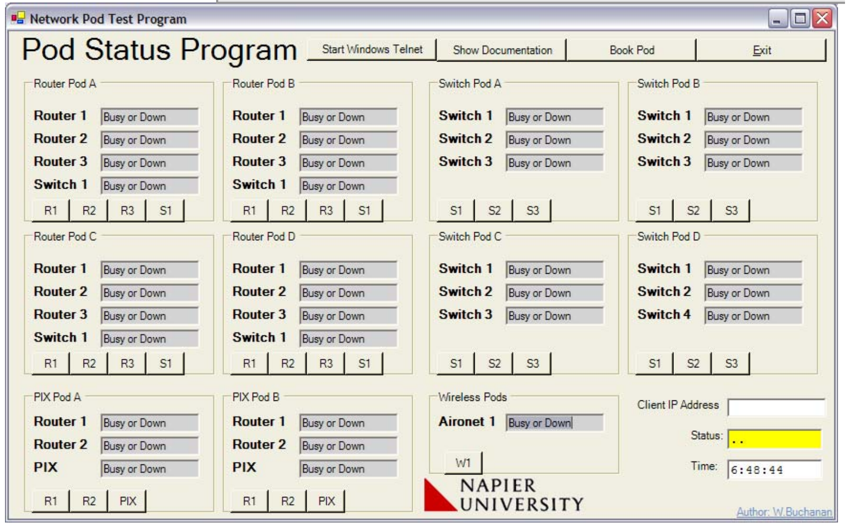

Windows Task Manager
File Options View Help
Applications Processes Performance Networking
Show processes from all users
End Process

Image Name	User Name	CPU	Mem Usage
explorer.exe	William Buchanan	00	12,800 K
wmplayer.exe	William Buchanan	00	3,604 K
realsched.exe	William Buchanan	00	120 K
WISPTIS.EXE	William Buchanan	00	728 K
taskmgr.exe	William Buchanan	00	4,820 K
msnmsgr.exe	William Buchanan	00	4,424 K
OD2State.exe	William Buchanan	00	712 K
sqlmangr.exe	William Buchanan	00	1,020 K
iproc488.exe	William Buchanan	00	424 K
mqtgsvc.exe	SYSTEM	00	256 K
iproc82357.exe	William Buchanan	00	244 K
iprocsvr.exe	William Buchanan	00	272 K
acrotray.exe	William Buchanan	00	736 K
mqsvc.exe	SYSTEM	00	844 K
CCEVTMGR.EXE	SYSTEM	00	428 K
WCESCOMM.EXE	William Buchanan	00	204 K
MSTORDB.EXE	William Buchanan	00	8,032 K
ctfmon.exe	William Buchanan	00	500 K
drst.exe	William Buchanan	00	2,536 K

ASP Web Page

Windows Application



Pod Status Program

Router Pod A
Router 1 Busy or Down
Router 2 Busy or Down
Router 3 Busy or Down
Switch 1 Busy or Down

Router Pod B
Router 1 Busy or Down
Router 2 Busy or Down
Router 3 Busy or Down
Switch 1 Busy or Down

Switch Pod A
Switch 1 Busy or Down
Switch 2 Busy or Down
Switch 3 Busy or Down

Switch Pod B
Switch 1 Busy or Down
Switch 2 Busy or Down
Switch 3 Busy or Down

Router Pod C
Router 1 Busy or Down
Router 2 Busy or Down
Router 3 Busy or Down
Switch 1 Busy or Down

Router Pod D
Router 1 Busy or Down
Router 2 Busy or Down
Router 3 Busy or Down
Switch 1 Busy or Down

Switch Pod C
Switch 1 Busy or Down
Switch 2 Busy or Down
Switch 3 Busy or Down

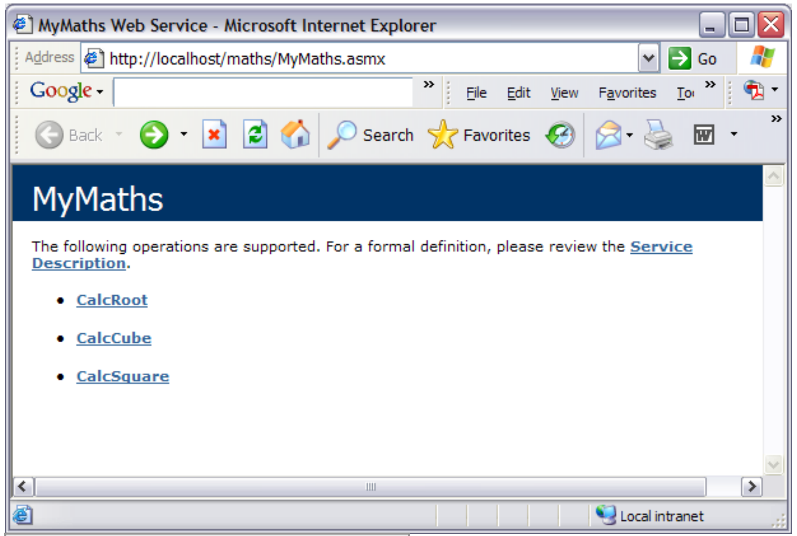
Switch Pod D
Switch 1 Busy or Down
Switch 2 Busy or Down
Switch 4 Busy or Down

PIX Pod A
Router 1 Busy or Down
Router 2 Busy or Down
PIX Busy or Down

PIX Pod B
Router 1 Busy or Down
Router 2 Busy or Down
PIX Busy or Down

Wireless Pods
Aironet 1 Busy or Down

Client IP Address
Status: . .
Time: 6:48:44

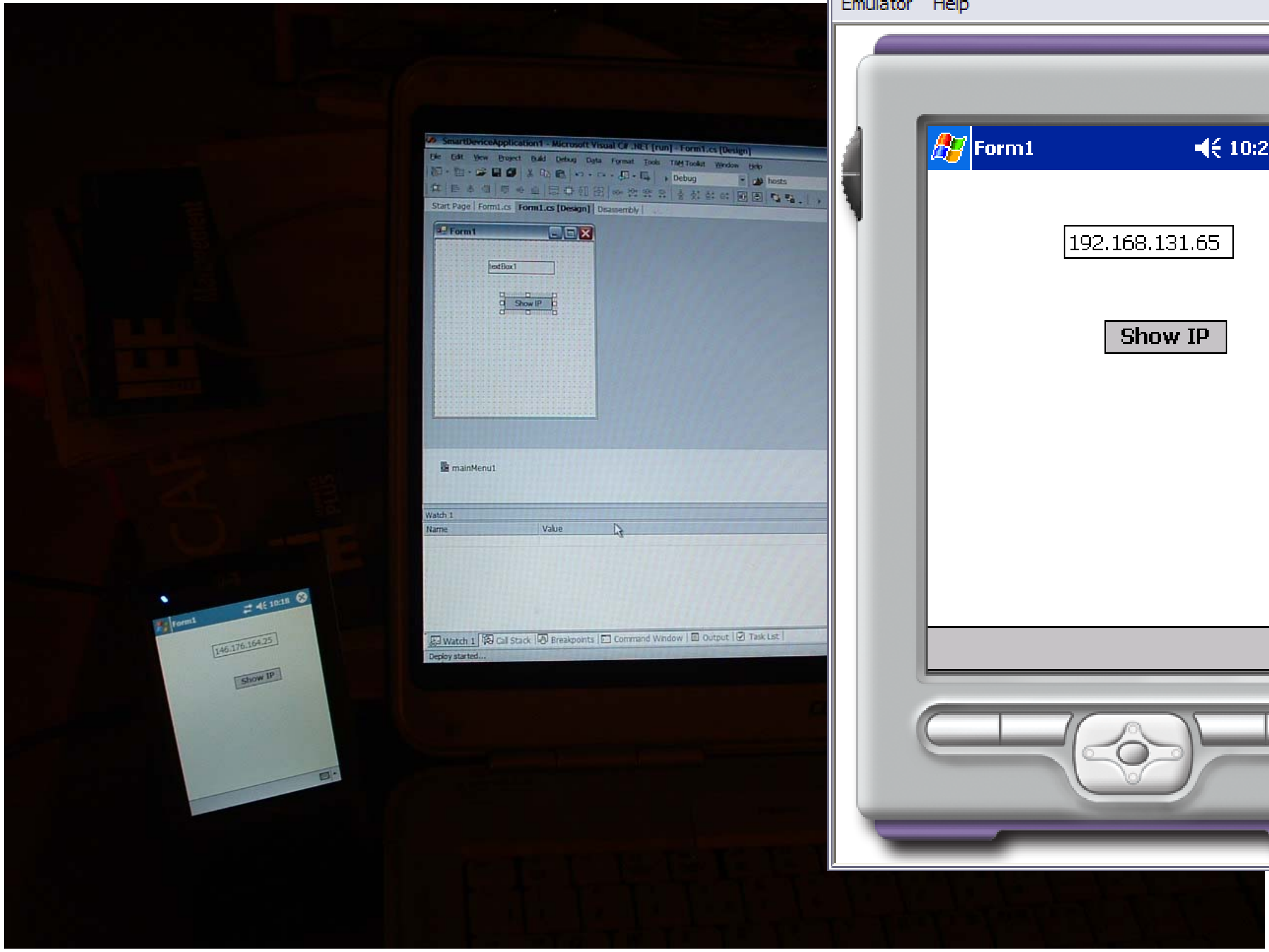

MyMaths Web Service - Microsoft Internet Explorer
Address http://localhost/maths/MyMaths.asmx
Go
Google
File Edit View Favorites Too
Back Search Favorites
Local intranet

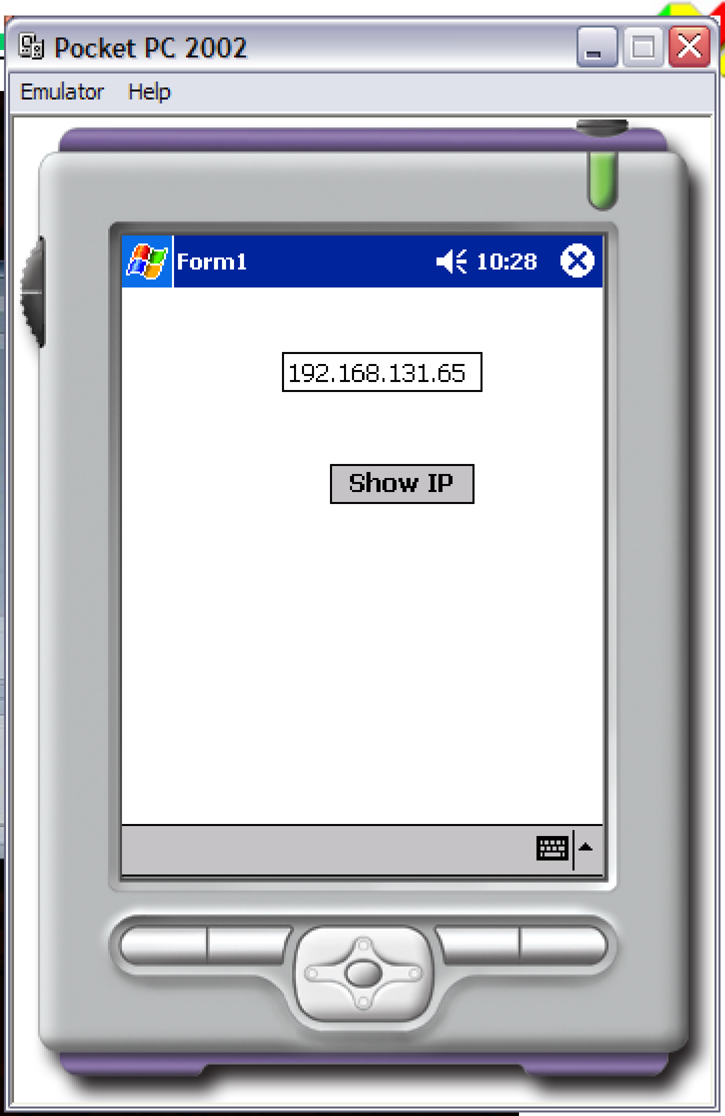
MyMaths

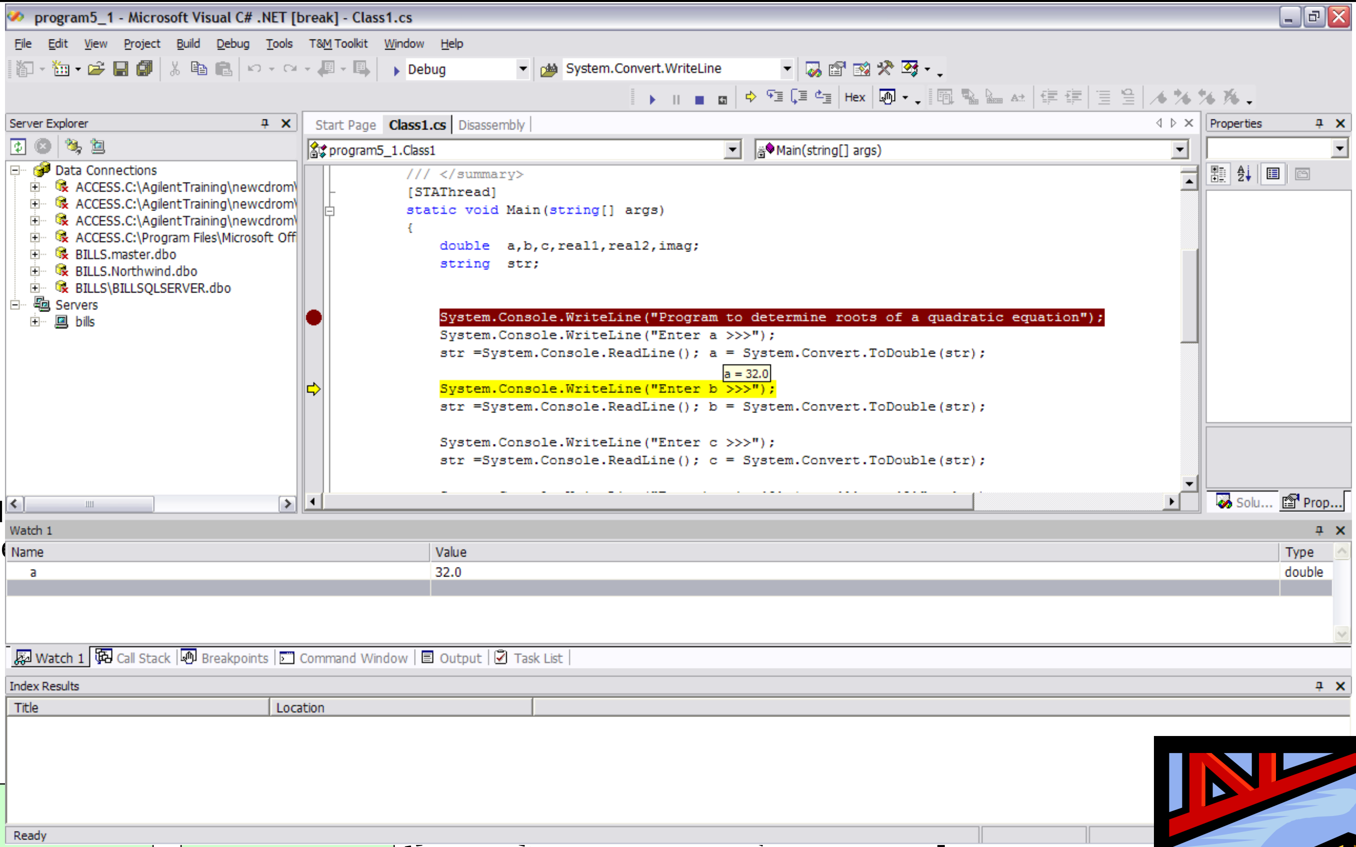
The following operations are supported. For a formal definition, please review the Service Description.

- CalcRoot
- CalcCube
- CalcSquare

Web Service


Form1
146.176.164.25
Show IP


Pocket PC 2002
Emulator
Help
Form1
10:28
192.168.131.65
Show IP

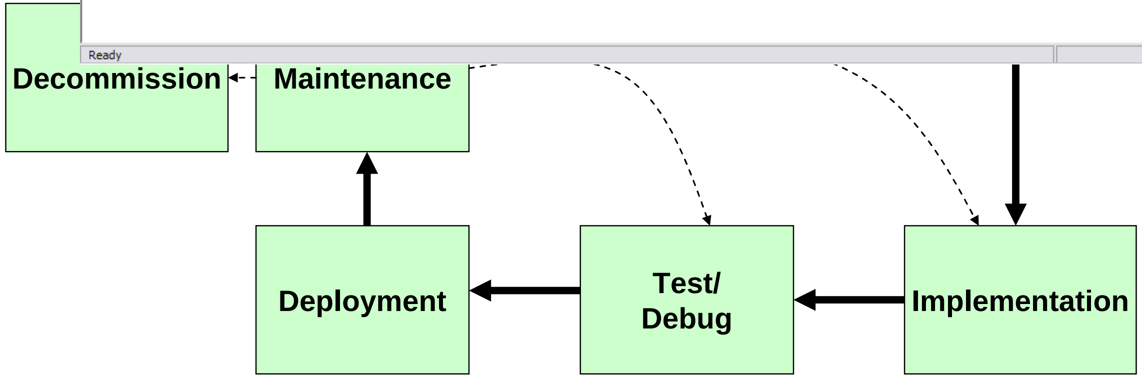


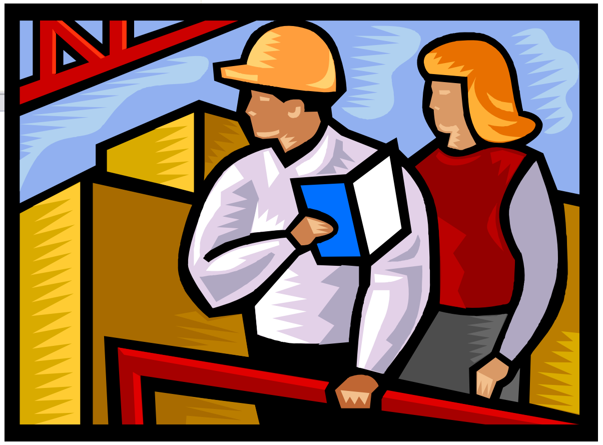
program5_1 - Microsoft Visual C# .NET [break] - Class1.cs

```
/// </summary>
[STAThread]
static void Main(string[] args)
{
    double  a,b,c,real1,real2,imag;
    string  str;


    System.Console.WriteLine("Program to determine roots of a quadratic equation");
    System.Console.WriteLine("Enter a >>>");
    str =System.Console.ReadLine(); a = System.Convert.ToDouble(str);
                                a = 32.0
    System.Console.WriteLine("Enter b >>>");
    str =System.Console.ReadLine(); b = System.Convert.ToDouble(str);

    System.Console.WriteLine("Enter c >>>");
    str =System.Console.ReadLine(); c = System.Convert.ToDouble(str);
```

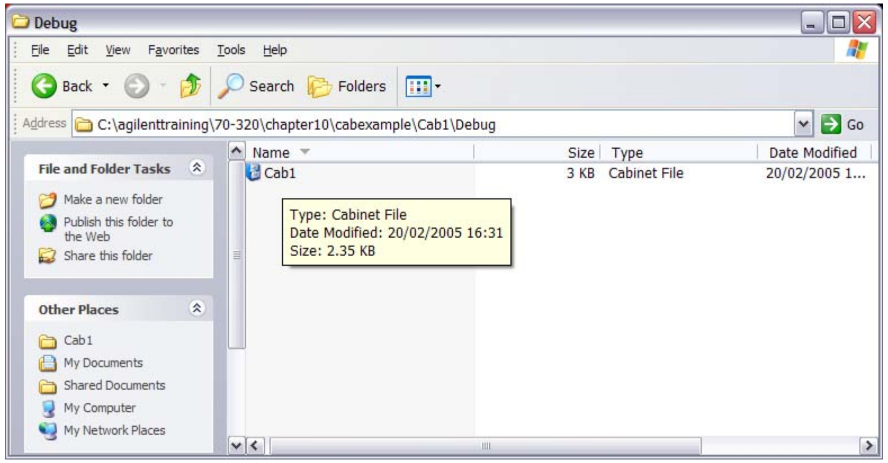
Watch 1

Name	Value	Type
a	32.0	double





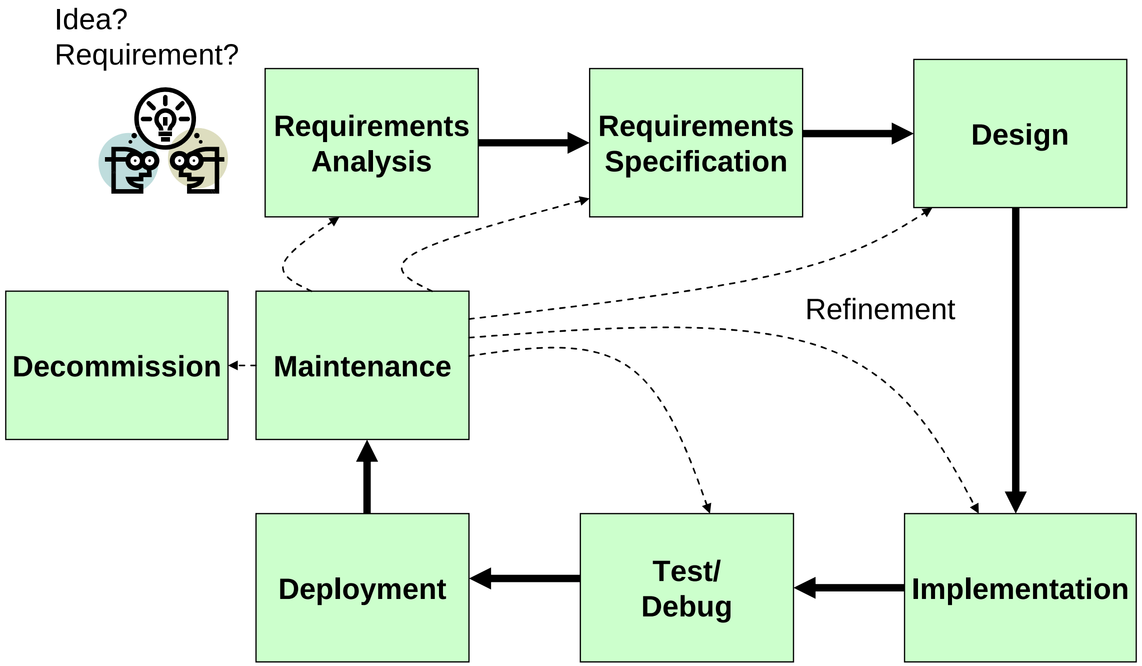
Debug/Test



Address: C:\agilenttraining\70-320\chapter10\cabexample\Cab1\Debug

Name	Size	Type	Date Modified
Cab1	3 KB	Cabinet File	20/02/2005 1...

Type: Cabinet File
Date Modified: 20/02/2005 16:31
Size: 2.35 KB

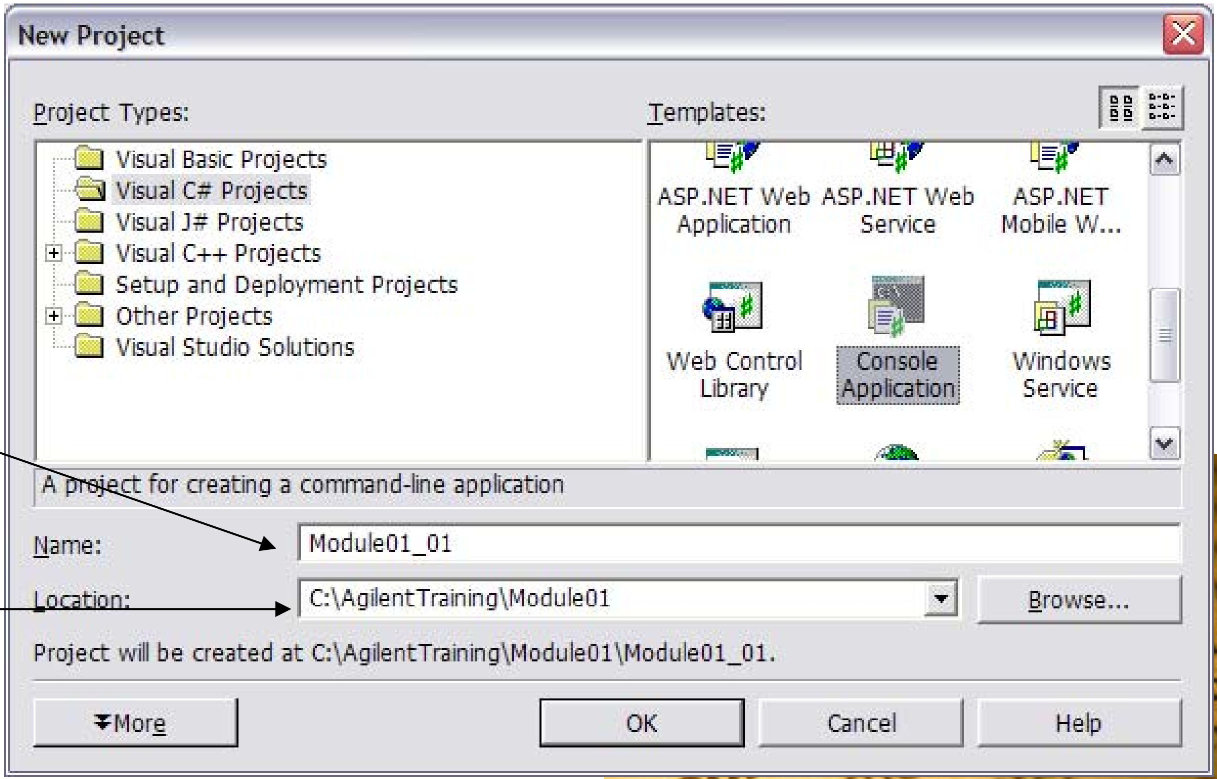


Enhanced Deployment:
- -CAB files.
- -XCOPY deployment.
- -MSI Installation.
- -Web deployment.

Deployment

Name of the folder
Which contains the
Project files
the project is stored.

Folder where
the project is stored.


New Project

Project Types:
- Visual Basic Projects
- Visual C# Projects
- Visual J# Projects
- Visual C++ Projects
- Setup and Deployment Projects
- Other Projects
- Visual Studio Solutions

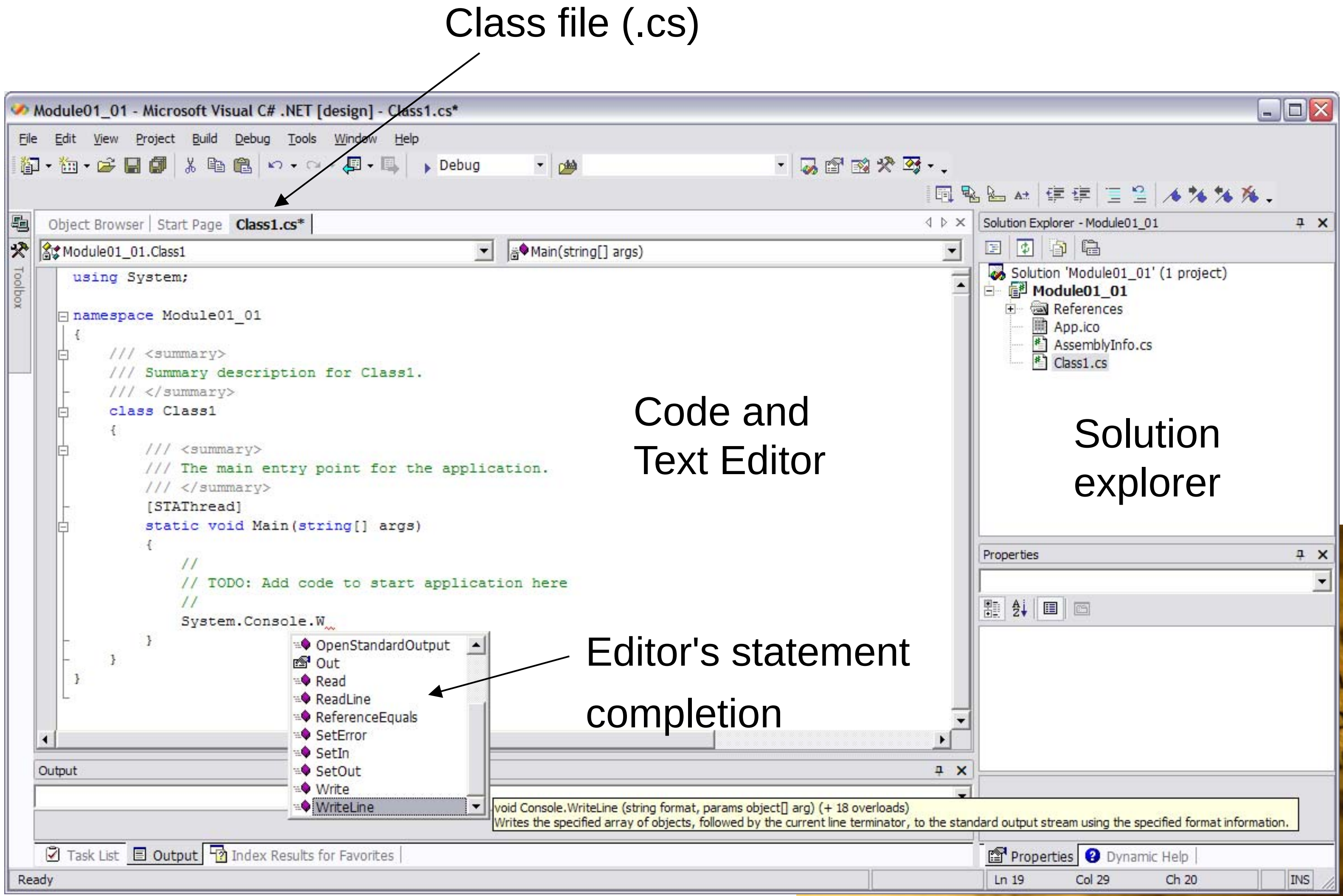
Templates:

A project for creating a command-line application

Name: Module01_01

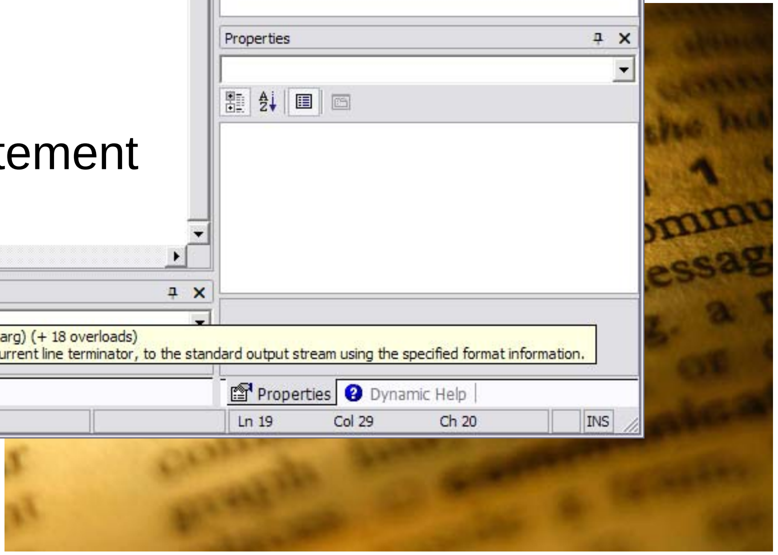
Location: C:\AgilentTraining\Module01

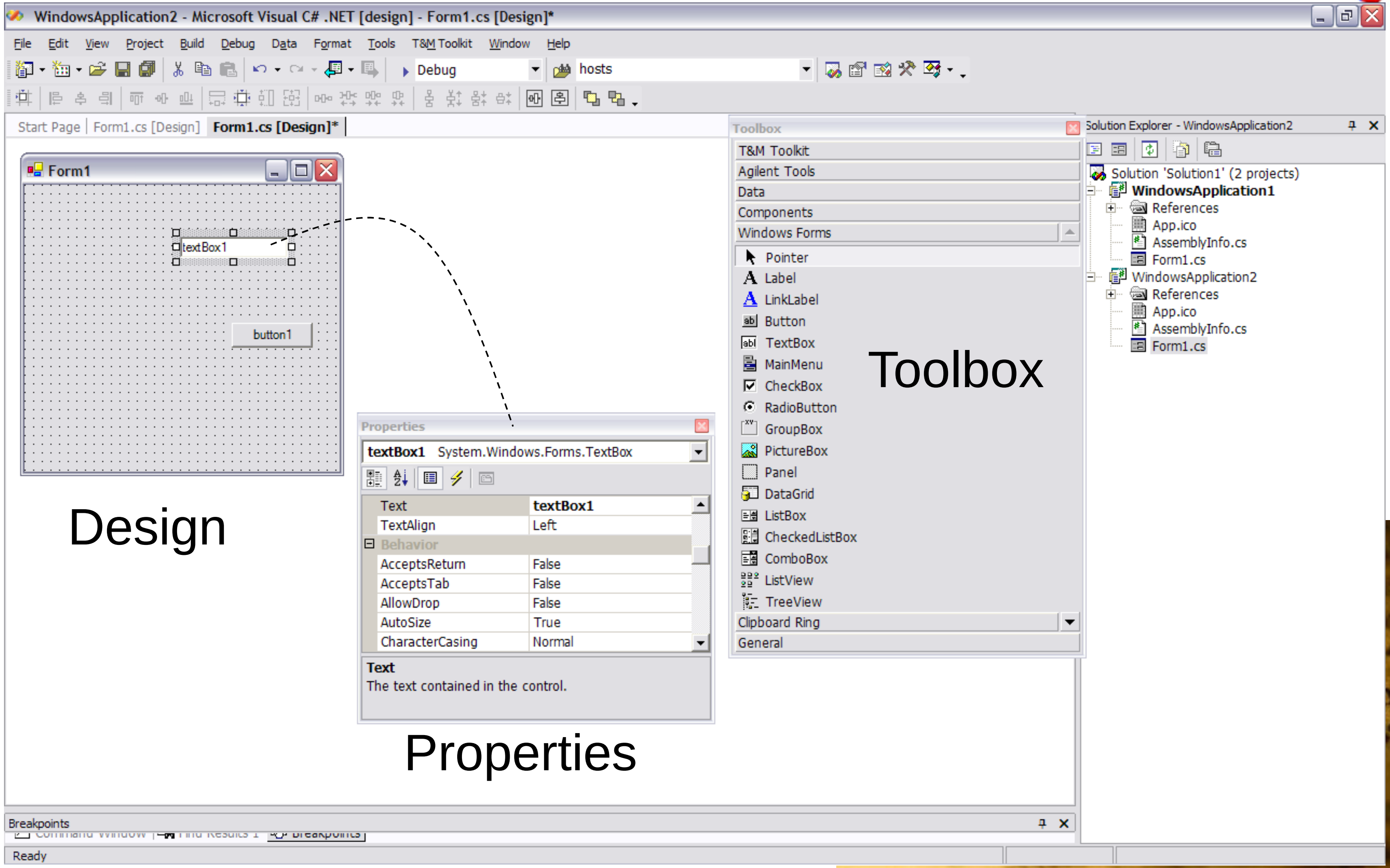
Project will be created at C:\AgilentTraining\Module01\Module01_01.



```
using System;

namespace Module01_01
{
    /// <summary>
    /// Summary description for Class1.
    /// </summary>
    class Class1
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main(string[] args)
        {
            //
            // TODO: Add code to start application here
            //
            System.Console.W
        }
    }
}
```

void Console.WriteLine (string format, params object[] arg) (+ 18 overloads)
Writes the specified array of objects, followed by the current line terminator, to the standard output stream using the specified format information.





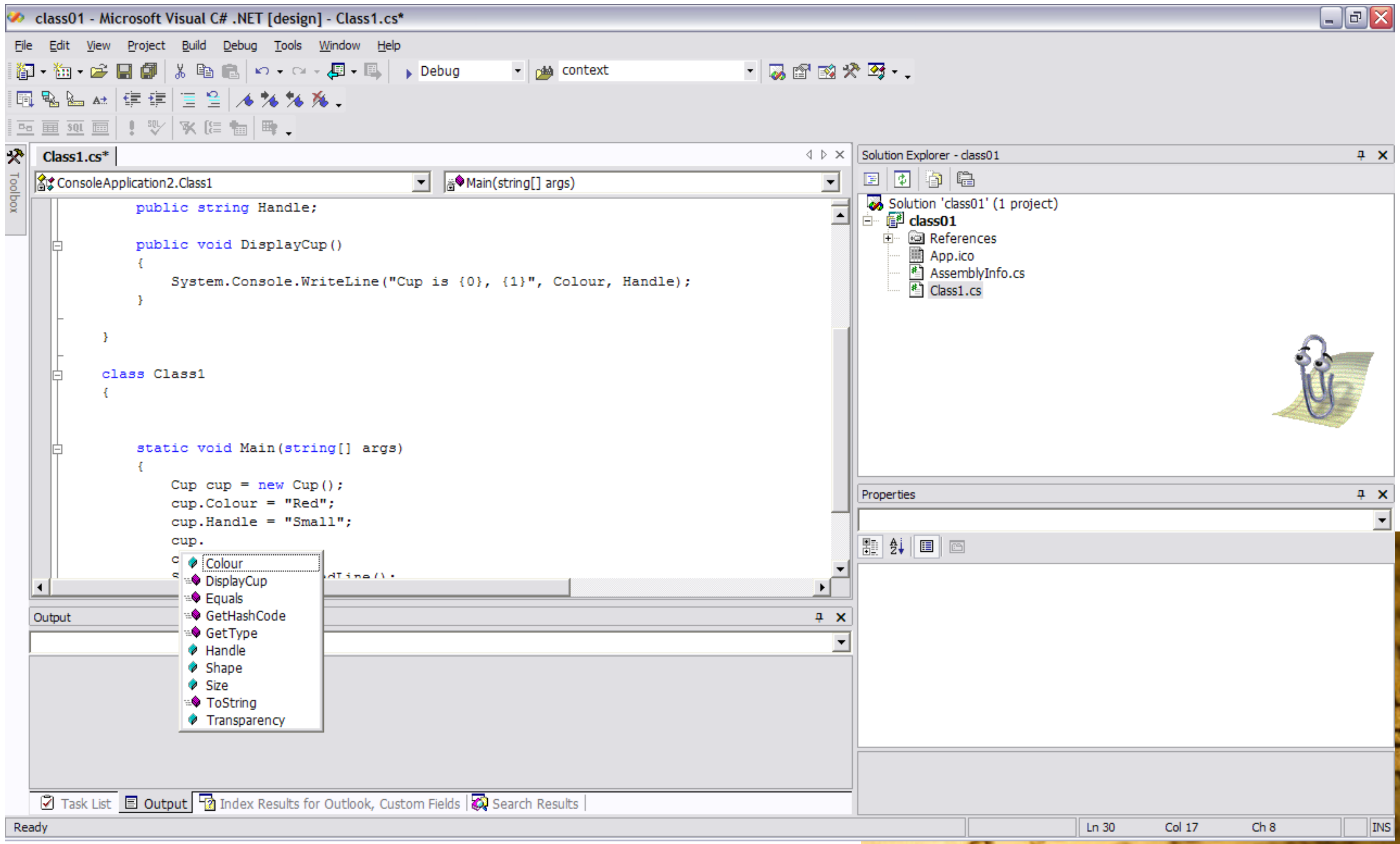
Text	textBox1
TextAlign	Left
Behavior	
AcceptsReturn	False
AcceptsTab	False
AllowDrop	False
AutoSize	True
CharacterCasing	Normal

Text
The text contained in the control.

Design

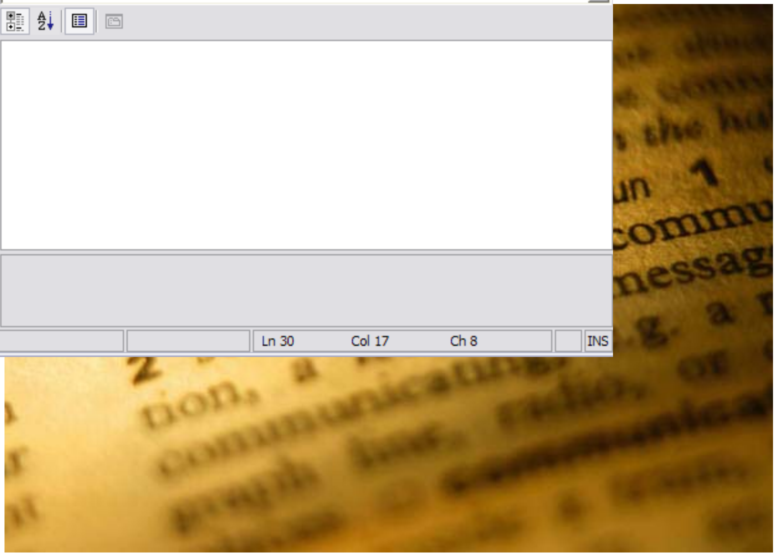
Toolbox

Properties

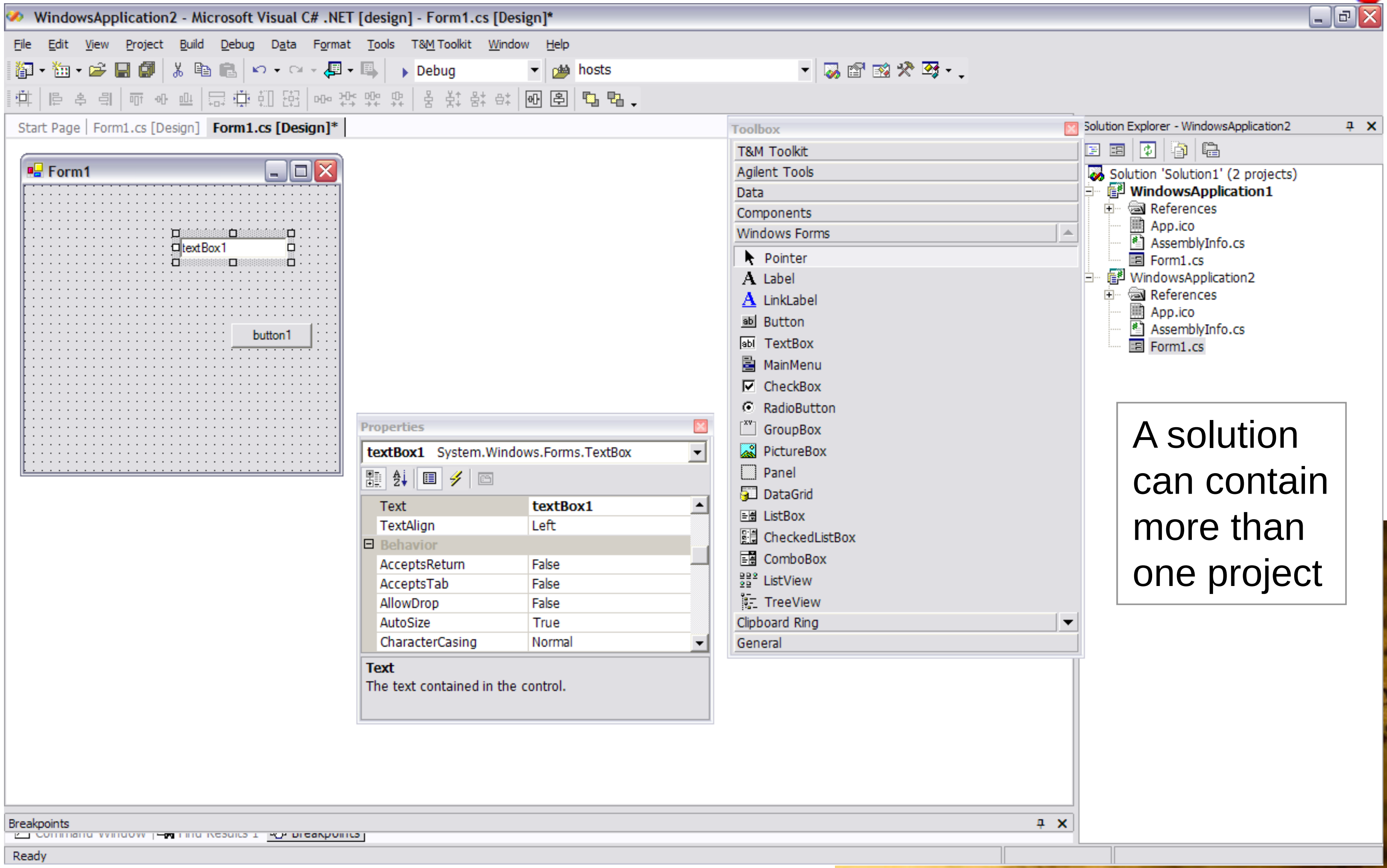

class01 - Microsoft Visual C# .NET [design] - Class1.cs*
File Edit View Project Build Debug Tools Window Help
Debug
context
Class1.cs*
ConsoleApplication2.Class1
Main(string[] args)
Solution Explorer - class01
Solution 'class01' (1 project)
class01
References
App.ico
AssemblyInfo.cs
Class1.cs
Properties
Output
Task List
Output
Index Results for Outlook, Custom Fields
Search Results
Ready
Ln 30
Col 17
Ch 8
INS

```
        public string Handle;

        public void DisplayCup()
        {
            System.Console.WriteLine("Cup is {0}, {1}", Colour, Handle);
        }

    }

    class Class1
    {


        static void Main(string[] args)
        {
            Cup cup = new Cup();
            cup.Colour = "Red";
            cup.Handle = "Small";
            cup.
            c
            S                dLine();
```

```
Colour
DisplayCup
Equals
GetHashCode
GetType
Handle
Shape
Size
ToString
Transparency
```



Example of Auto-Complete



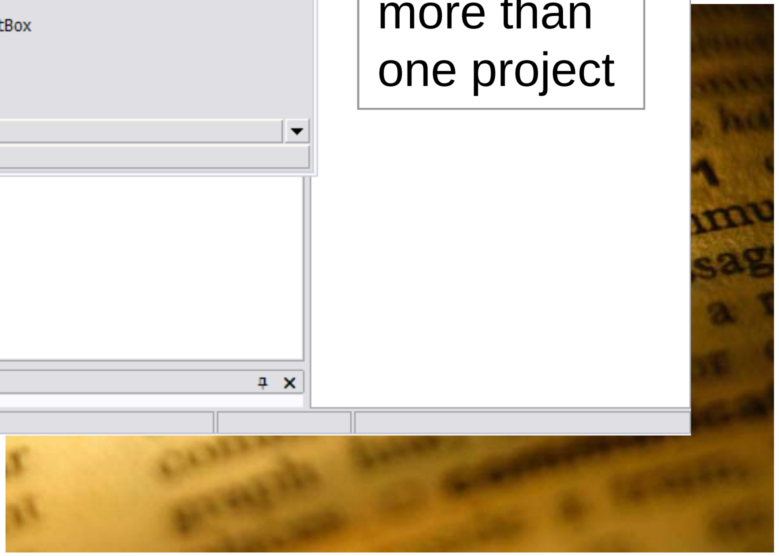
Properties

textBox1 System.Windows.Forms.TextBox

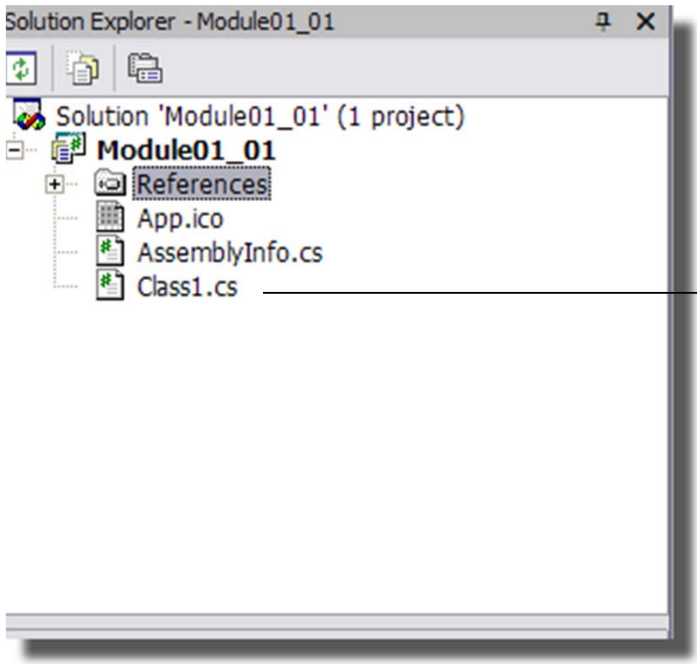
Text	textBox1
TextAlign	Left
Behavior	
AcceptsReturn	False
AcceptsTab	False
AllowDrop	False
AutoSize	True
CharacterCasing	Normal

Text
The text contained in the control.

A solution can contain more than one project



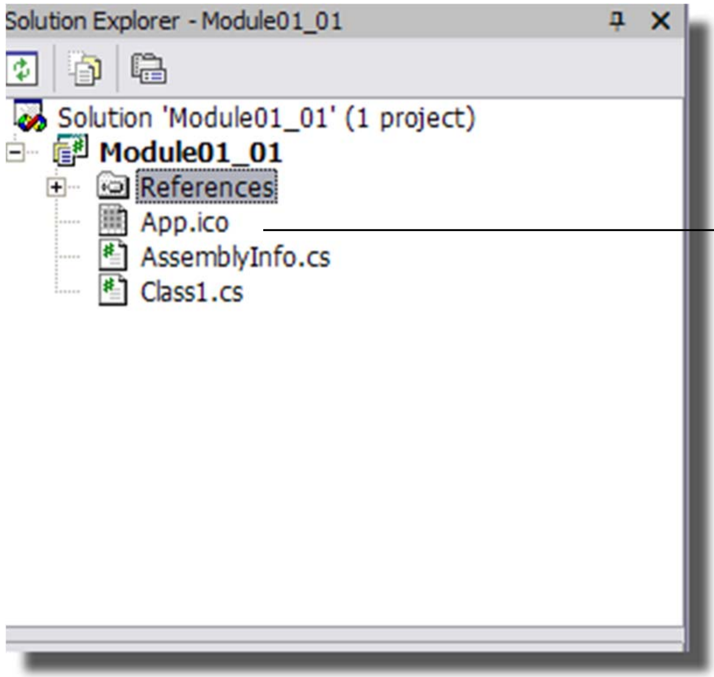
Solution File (.sln)


Solution Explorer - Module01_01
Solution 'Module01_01' (1 project)
Module01_01
References
App.ico
AssemblyInfo.cs
Class1.cs

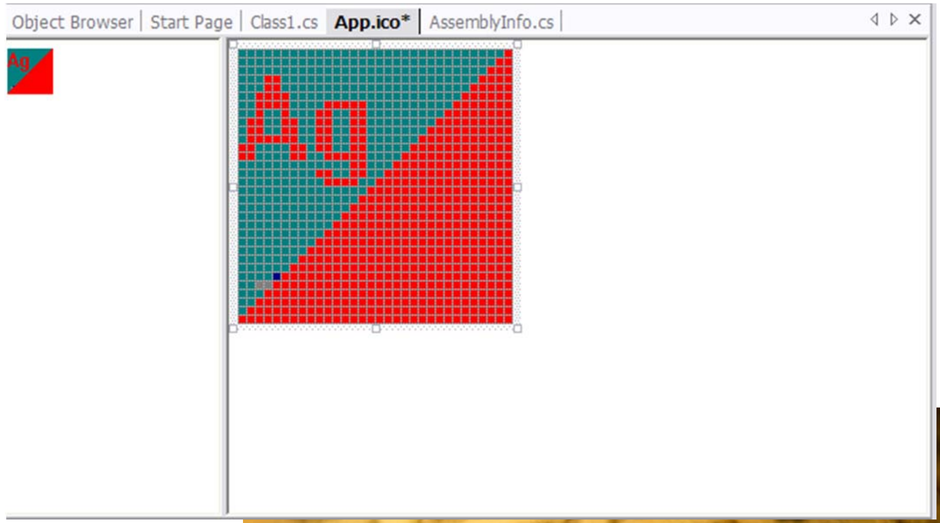
Class1.cs

```
using System;

namespace Module01_01
{
     /// <summary>
     /// Summary description for Class1.
     /// </summary>
   class Class1
   {
     /// <summary>
     /// The main entry point for the application.
     /// </summary>
     static void Main(string[] args)
     {
     //
     // TODO: Add code to start application here
     System.Console.WriteLine(“SoC Course");
     System.Console.ReadLine();
     }
   }
}
```

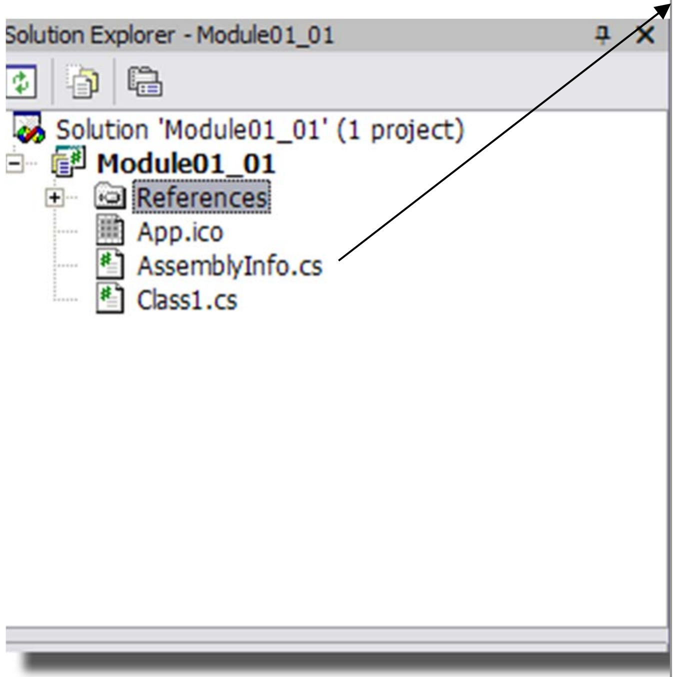
Solution File (.sln)


Solution Explorer - Module01_01

Solution 'Module01_01' (1 project)
Module01_01
References
App.ico
AssemblyInfo.cs
Class1.cs



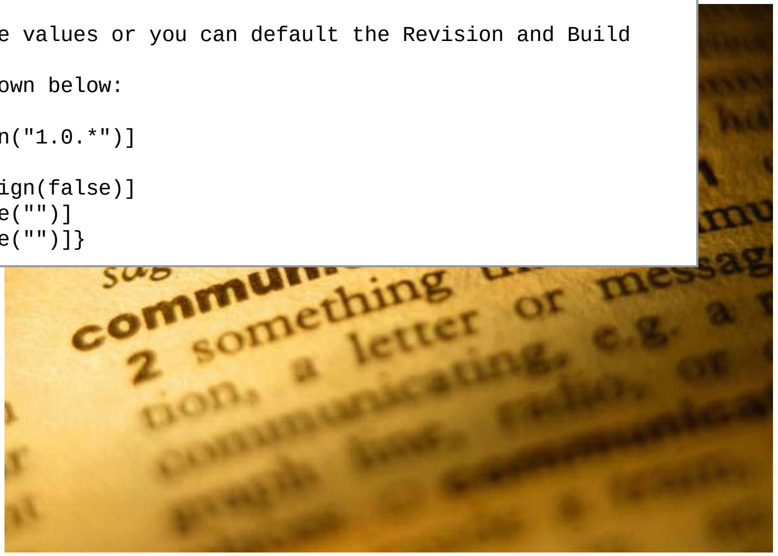
ICO file


Solution Explorer - Module01_01
Solution 'Module01_01' (1 project)
Module01_01
References
App.ico
AssemblyInfo.cs
Class1.cs

AssemblyInfo.cs

```
using System.Reflection;
using System.Runtime.CompilerServices;

//
// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
//
[assembly: AssemblyTitle("")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("")]
[assembly: AssemblyCopyright("")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]

//
// Version information for an assembly consists of the following four values:
//
//      Major Version
//      Minor Version
//      Build Number
//      Revision
//
// You can specify all the values or you can default the Revision and Build
      Numbers
// by using the '*' as shown below:

[assembly: AssemblyVersion("1.0.*")]

[assembly: AssemblyDelaySign(false)]
[assembly: AssemblyKeyFile("")]
[assembly: AssemblyKeyName("")]}
```

.NET uses assembly to represent a single unit. An assembly is a collection of files that appear as a single unit, such as a single DLL or an EXE.

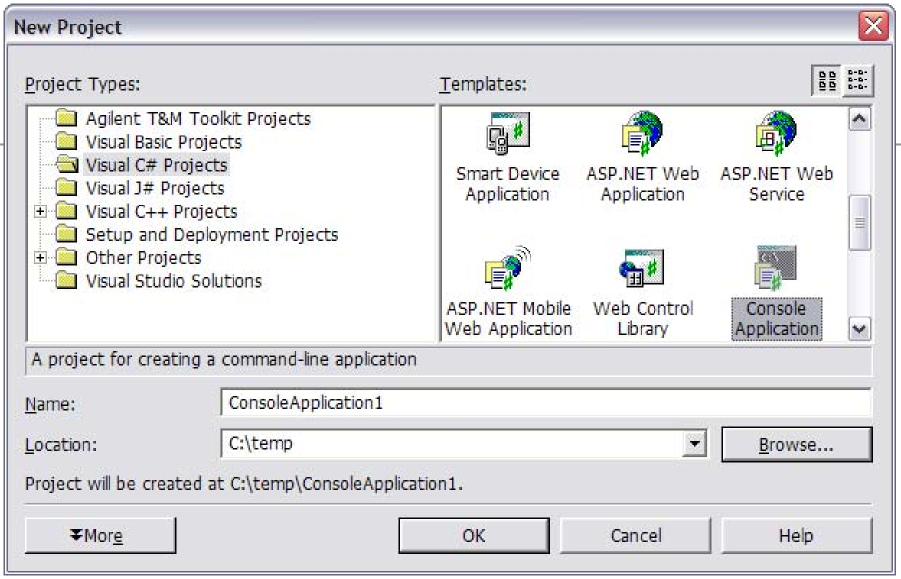


Solution File (.sln)

Simple Console Application

Bill Buchanan

```
using System;
namespace ConsoleApplication1
{
  class Class1
  {
     static void Main(string[] args)
     {
       System.Console.WriteLine("This is my first program");
       System.Console.ReadLine();
     }
  }
}
```


New Project

Project Types:
- Agilent T&M Toolkit Projects
- Visual Basic Projects
- Visual C# Projects
- Visual J# Projects
- Visual C++ Projects
- Setup and Deployment Projects
- Other Projects
- Visual Studio Solutions

Templates:
- Smart Device Application
- ASP.NET Web Application
- ASP.NET Web Service
- ASP.NET Mobile Web Application
- Web Control Library
- Console Application

A project for creating a command-line application

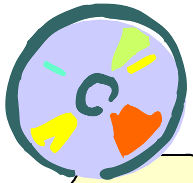
Name: ConsoleApplication1

Location: C:\temp

Project will be created at C:\temp\ConsoleApplication1.

Tutorial Session 1:

Q1.1 and Q1.2

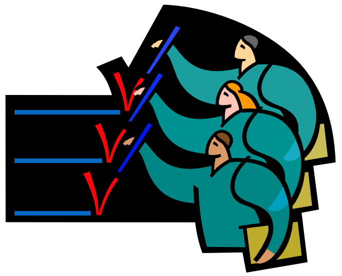


Presentations

Notes

SourceCode

Tutorials



An Introduction to Object-Orientation

Bill Buchanan

Some Cups

Parameter	Cup 1	Cup 2	Cup3
Shape (Standard/Square/Mug)	Standard	Square	Mug
Colour (Red/Blue/Green)	Blue	Red	Green
Size (Small/Medium/Large)	Small	Large	Small
Transparency (0 to 100%)	100%	50%	25%
Handle type (Small/Large)	Small	Small	Large

In object-orientation:A collection of parameters defines a **class**.

Class for the cup is thus: **Shape**, **Colour**, **Size**, **Transparency**, **HandleType**.

In object-orientation: Objects are created from classes.

A Quick Introduction to Object-Orientation

```
using System;

namespace ConsoleApplication2
{

     public class Cup
     {
          public string Shape;
          public string Colour;
          public string Size;
          public int Transparency;
          public string Handle;

          public void DisplayCup()
          {
              System.Console.WriteLine("Cup is {0}, {1}", Colour, Handle);
          }

     }

     class Class1
     {
          static void Main(string[] args)
          {
              Cup cup = new Cup();
              cup.Colour = "Red";
              cup.Handle = "Small";
              cup.DisplayCup();
              System.Console.ReadLine();
          }
     }
}
```

Class definitions

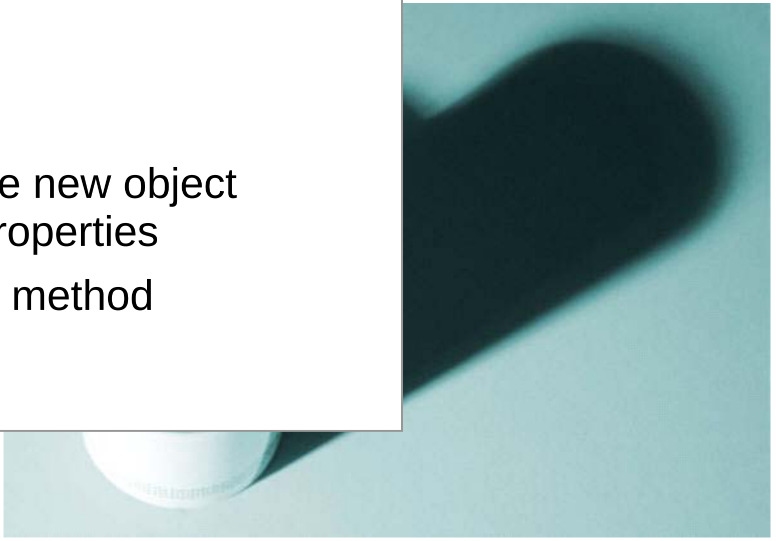
Available variables (properties)

Method

Create new object

Set properties

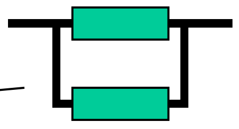
Apply method

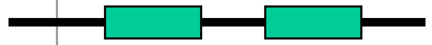


Example C# Program using Object-Orientation

```
using System;

namespace ConsoleApplication2
{
     public class Circuit
     {
          public double Parallel(double r1, double r2)
          {
               return((r1*r2)/(r1+r2));
          }
          public double Series(double r1, double r2)
          {
               return(r1+r2);
          }
     }

     class Class1
     {
          static void Main(string[] args)
          {
               double v1=100,v2=100;
               double res;

               Circuit cir = new Circuit();

               res=cir.Parallel(v1,v2);
               System.Console.WriteLine("Parallel resistance is {0} ohms",res);

               res=cir.Series(100,100);
               System.Console.WriteLine("Series resistance is {0} ohms",res);

               System.Console.ReadLine();
          }
     }
}
```

Class definitions





Another example

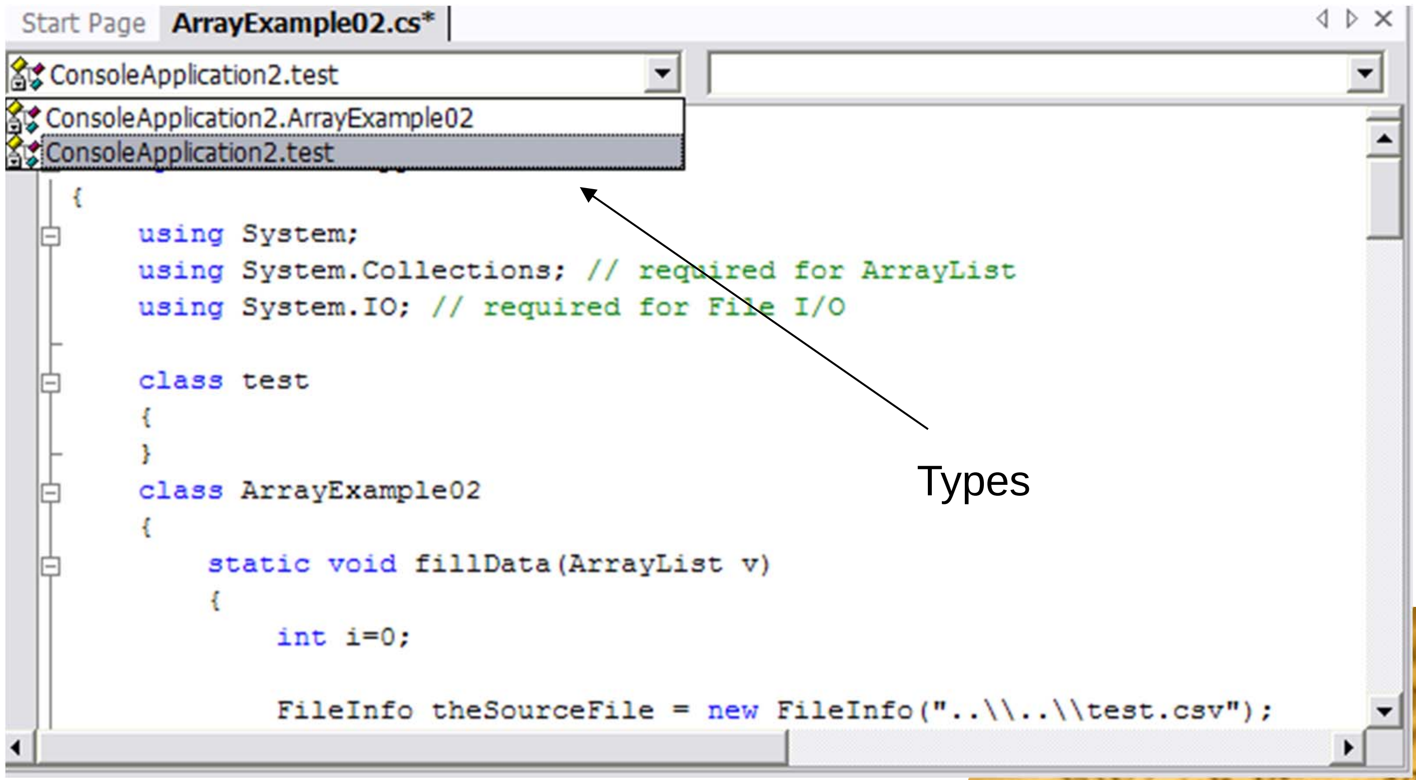
```
using System;
namespace ConsoleApplication2
{
   public class Complex
   {
      public double real;
      public double imag;

      public double mag()
      {
         return (Math.Sqrt(real*real+imag*imag));
      }
      public double angle()
      {
         return (Math.Atan(imag/real)*180/Math.PI);
      }
   }
   class Class1
   {
      static void Main(string[] args)
      {
         string str;
         double mag,angle;
         Complex r = new Complex();

         System.Console.Write("Enter real value >>");
         str=System.Console.ReadLine();
         r.real = Convert.ToInt32(str);

         System.Console.Write("Enter imag value >>");
         str=System.Console.ReadLine();
         r.imag = Convert.ToInt32(str);

         mag=r.mag();
         angle=r.angle();

         System.Console.WriteLine("Mag is {0} and angle is {1}",mag,angle);
         System.Console.ReadLine();
      }
   }
}
```

$$z = x + jy$$

$$|z| = \sqrt{x^2 + y^2}$$

$$\langle z \rangle = \tan^{-1}\left(\frac{y}{x}\right)$$

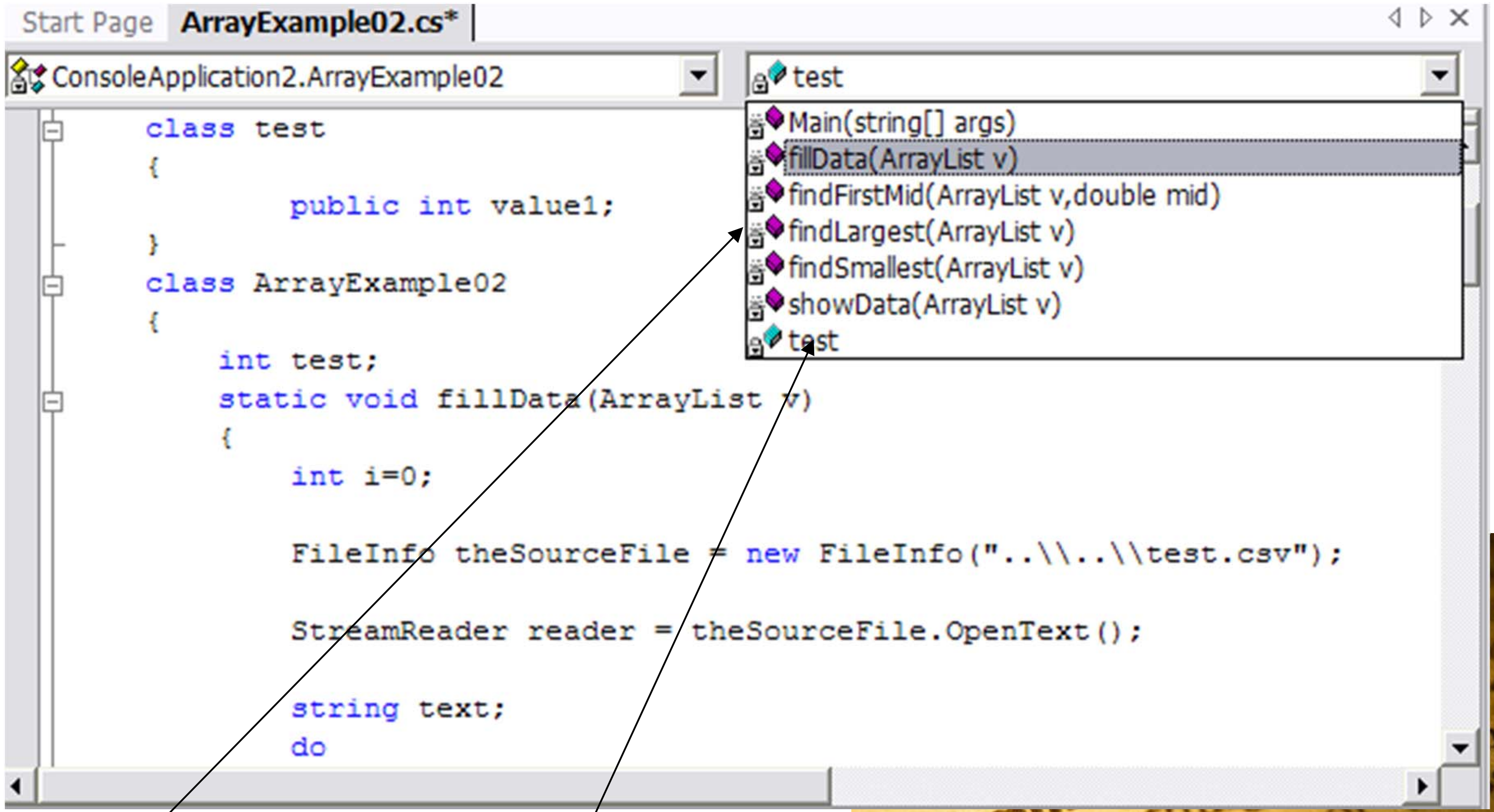
Another example



```
{
    using System;
    using System.Collections; // required for ArrayList
    using System.IO; // required for File I/O

    class test
    {
    }
    class ArrayExample02
    {
        static void fillData(ArrayList v)
        {
            int i=0;

            FileInfo theSourceFile = new FileInfo("..\\..\\test.csv");
```

Namespace: ConsoleAppplication2

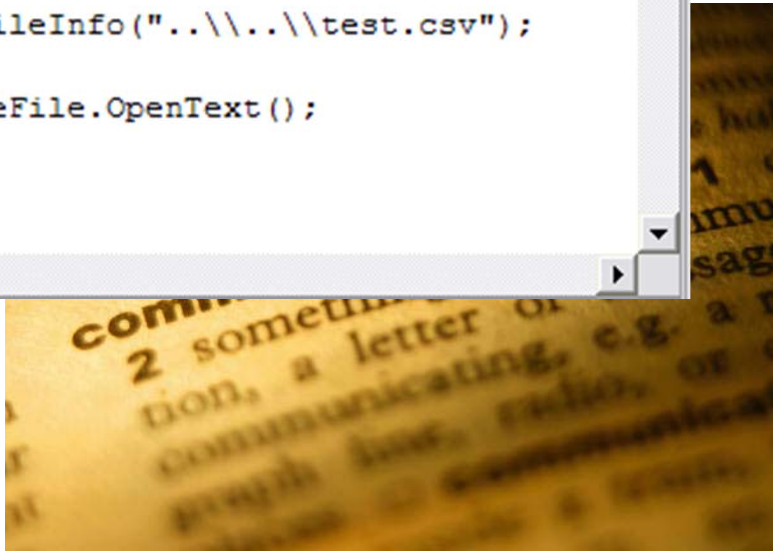
Types



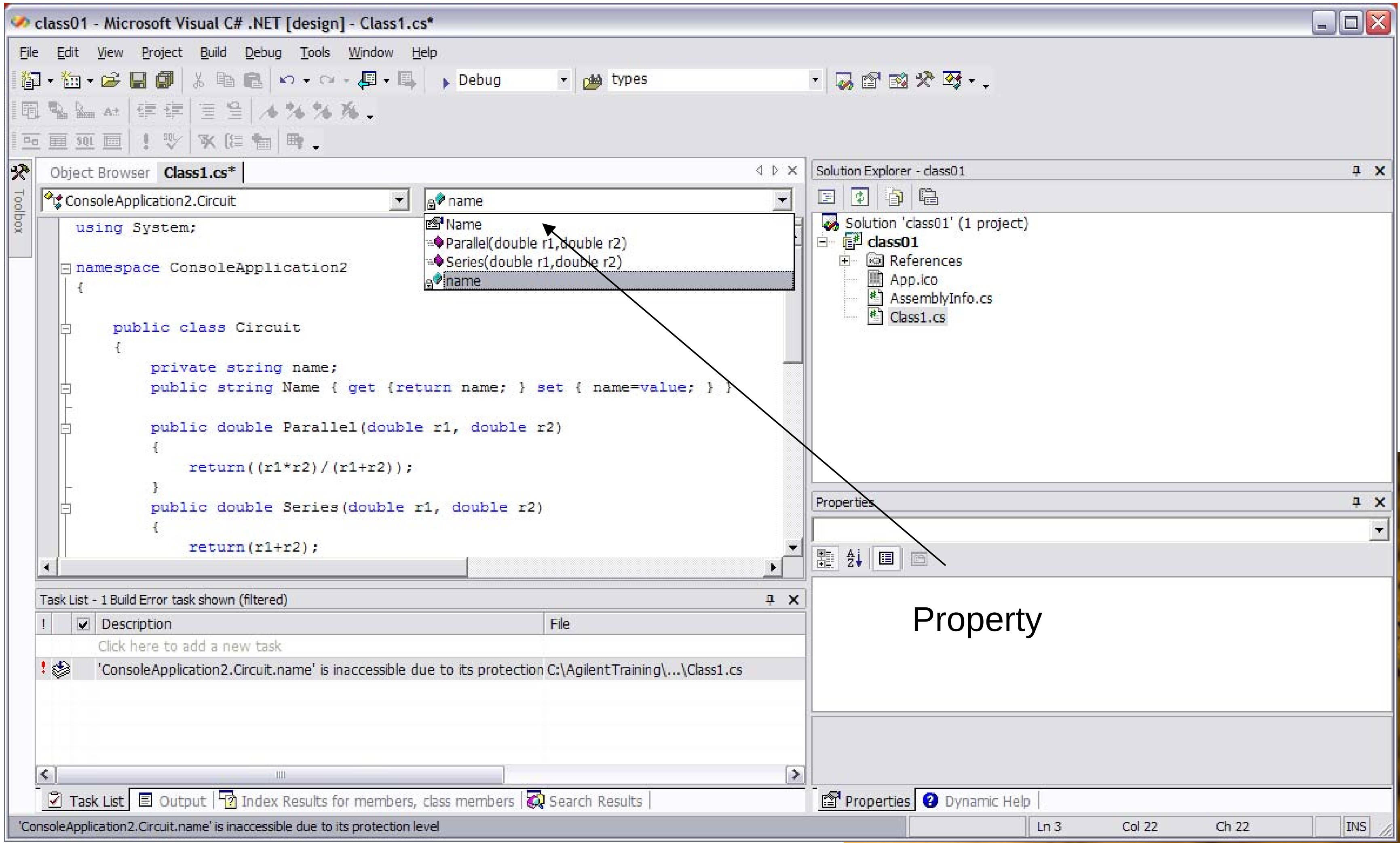
```
class test
{
        public int value1;
}
class ArrayExample02
{
    int test;
    static void fillData(ArrayList v)
    {
        int i=0;

        FileInfo theSourceFile = new FileInfo("..\\..\\test.csv");

        StreamReader reader = theSourceFile.OpenText();

        string text;
        do
```

Methods

Variable



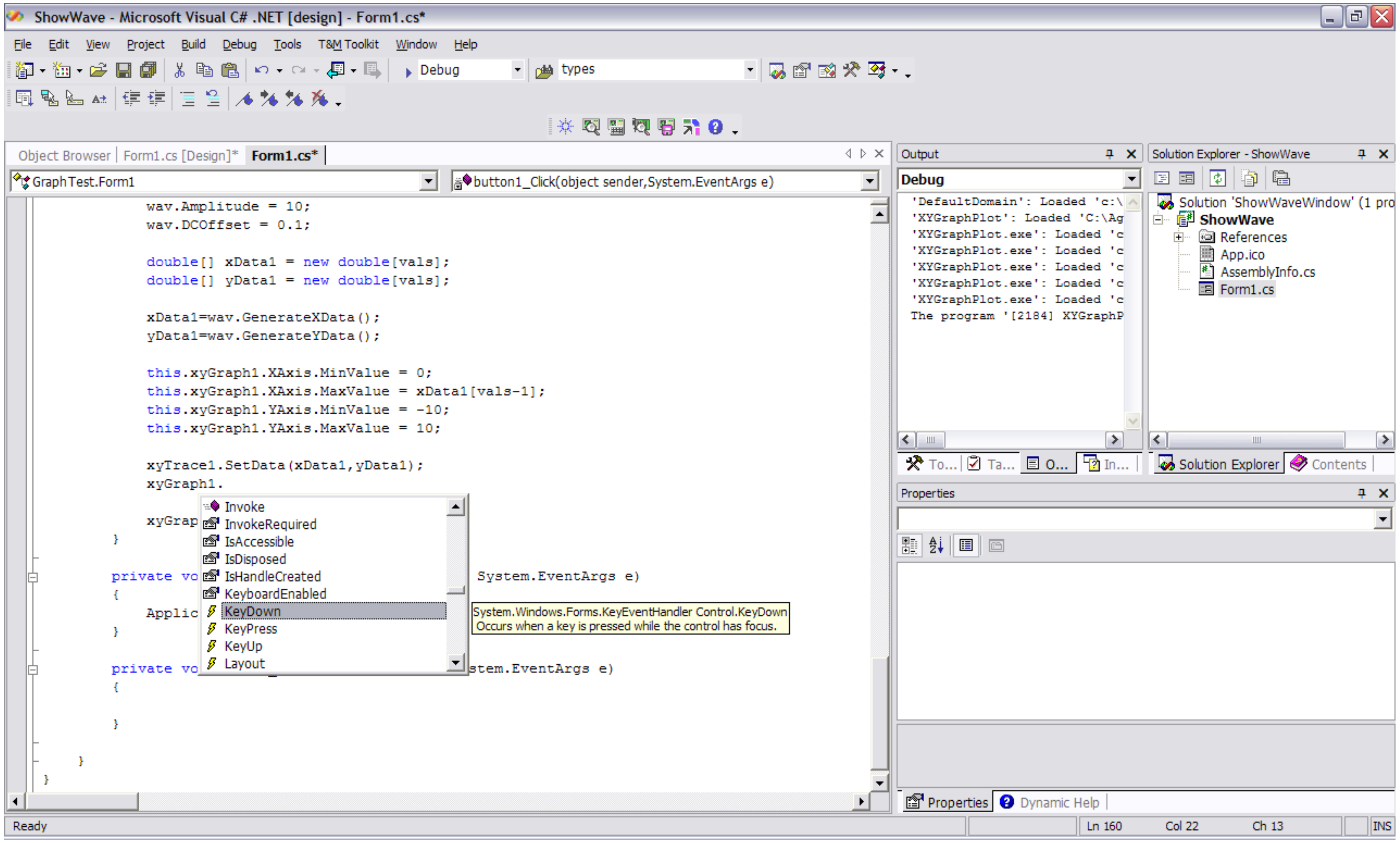
Members


class01 - Microsoft Visual C# .NET [design] - Class1.cs*
File Edit View Project Build Debug Tools Window Help
Debug
types
Object Browser
Class1.cs*
ConsoleApplication2.Circuit
name
Name
Parallel(double r1,double r2)
Series(double r1,double r2)
name
Solution Explorer - class01
Solution 'class01' (1 project)
class01
References
App.ico
AssemblyInfo.cs
Class1.cs
Properties
Task List - 1 Build Error task shown (filtered)
Description
File
Click here to add a new task
Task List
Output
Index Results for members, class members
Search Results
Properties
Dynamic Help
Ln 3
Col 22
Ch 22
INS

```
using System;

namespace ConsoleApplication2
{

    public class Circuit
    {
        private string name;
        public string Name { get {return name; } set { name=value; } }

        public double Parallel(double r1, double r2)
        {
            return((r1*r2)/(r1+r2));
        }
        public double Series(double r1, double r2)
        {
            return(r1+r2);
```

```
'ConsoleApplication2.Circuit.name' is inaccessible due to its protection C:\AgilentTraining\...\Class1.cs
'ConsoleApplication2.Circuit.name' is inaccessible due to its protection level
```

Property

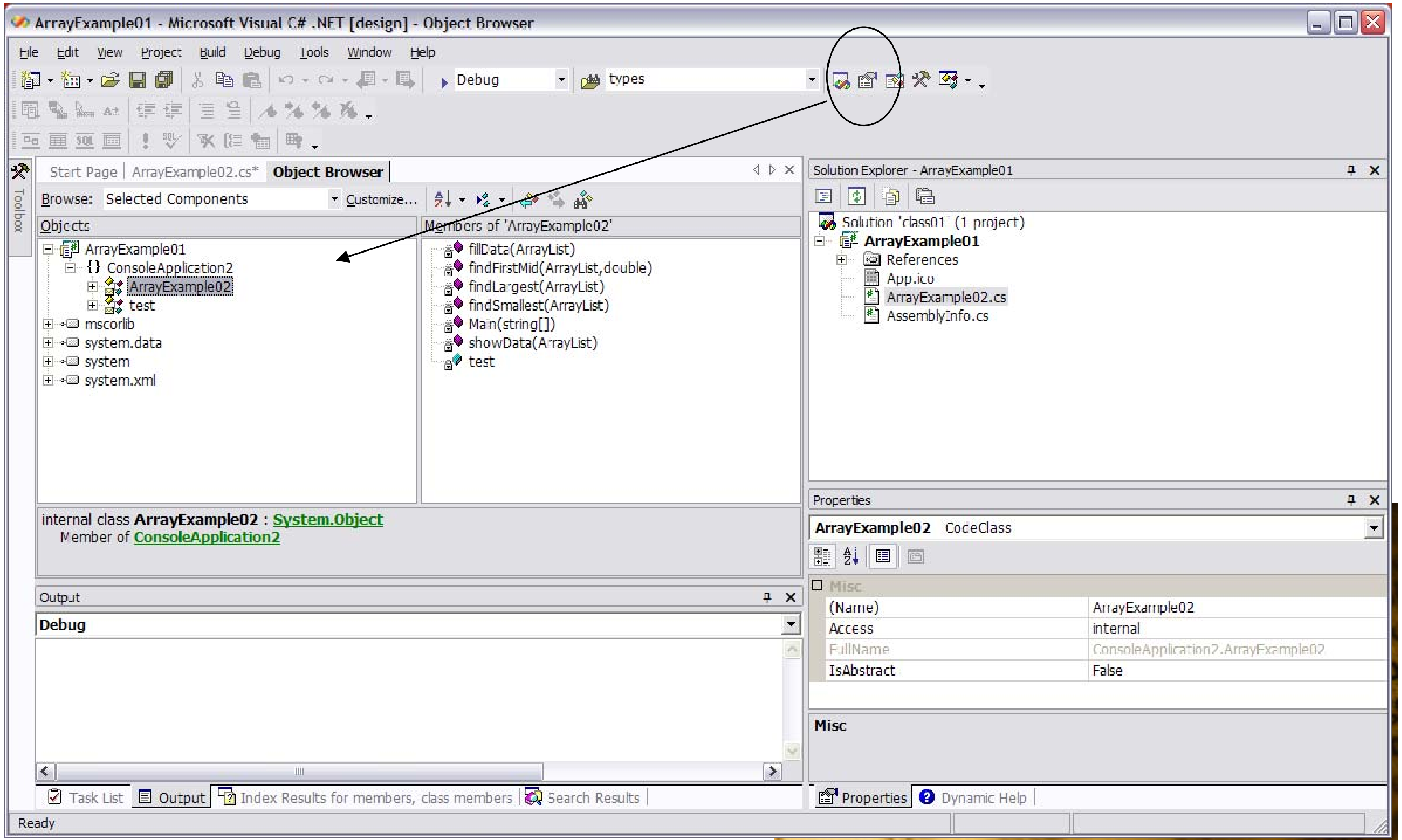

ShowWave - Microsoft Visual C# .NET [design] - Form1.cs*
File Edit View Project Build Debug Tools T&M Toolkit Window Help
Debug
types
Object Browser
Form1.cs [Design]*
Form1.cs*
GraphTest.Form1
button1_Click(object sender,System.EventArgs e)
Output
Debug
Solution Explorer - ShowWave
Solution 'ShowWaveWindow' (1 pro
ShowWave
References
App.ico
AssemblyInfo.cs
Form1.cs
To...
Ta...
O...
In...
Solution Explorer
Contents
Properties
Properties
Dynamic Help
Ready
Ln 160
Col 22
Ch 13
INS

```
            wav.Amplitude = 10;
            wav.DCOffset = 0.1;

            double[] xData1 = new double[vals];
            double[] yData1 = new double[vals];

            xData1=wav.GenerateXData();
            yData1=wav.GenerateYData();

            this.xyGraph1.XAxis.MinValue = 0;
            this.xyGraph1.XAxis.MaxValue = xData1[vals-1];
            this.xyGraph1.YAxis.MinValue = -10;
            this.xyGraph1.YAxis.MaxValue = 10;

            xyTrace1.SetData(xData1,yData1);
            xyGraph1.

            xyGrap
        }

        private vo                            System.EventArgs e)
        {
            Applic
        }

        private vo                           stem.EventArgs e)
        {

        }

    }
}
```

```
Invoke
InvokeRequired
IsAccessible
IsDisposed
IsHandleCreated
KeyboardEnabled
KeyDown
KeyPress
KeyUp
Layout
```

```
System.Windows.Forms.KeyEventHandler Control.KeyDown
Occurs when a key is pressed while the control has focus.
```

```
'DefaultDomain': Loaded 'c:\
'XYGraphPlot': Loaded 'C:\Ag
'XYGraphPlot.exe': Loaded 'c
'XYGraphPlot.exe': Loaded 'c
'XYGraphPlot.exe': Loaded 'c
'XYGraphPlot.exe': Loaded 'c
'XYGraphPlot.exe': Loaded 'c
The program '[2184] XYGraphP
```


ArrayExample01 - Microsoft Visual C# .NET [design] - Object Browser
File Edit View Project Build Debug Tools Window Help
Debug
types
Start Page
ArrayExample02.cs*
Object Browser
Browse: Selected Components
Customize...
Objects
Members of 'ArrayExample02'
Solution Explorer - ArrayExample01
Properties
Output
Task List
Output
Index Results for members, class members
Search Results
Dynamic Help
Ready

ArrayExample01
{} ConsoleApplication2
ArrayExample02
test
mscorlib
system.data
system
system.xml

fillData(ArrayList)
findFirstMid(ArrayList,double)
findLargest(ArrayList)
findSmallest(ArrayList)
Main(string[])
showData(ArrayList)
test

internal class **ArrayExample02** : System.Object
Member of ConsoleApplication2

Solution 'class01' (1 project)
ArrayExample01
References
App.ico
ArrayExample02.cs
AssemblyInfo.cs

ArrayExample02 CodeClass

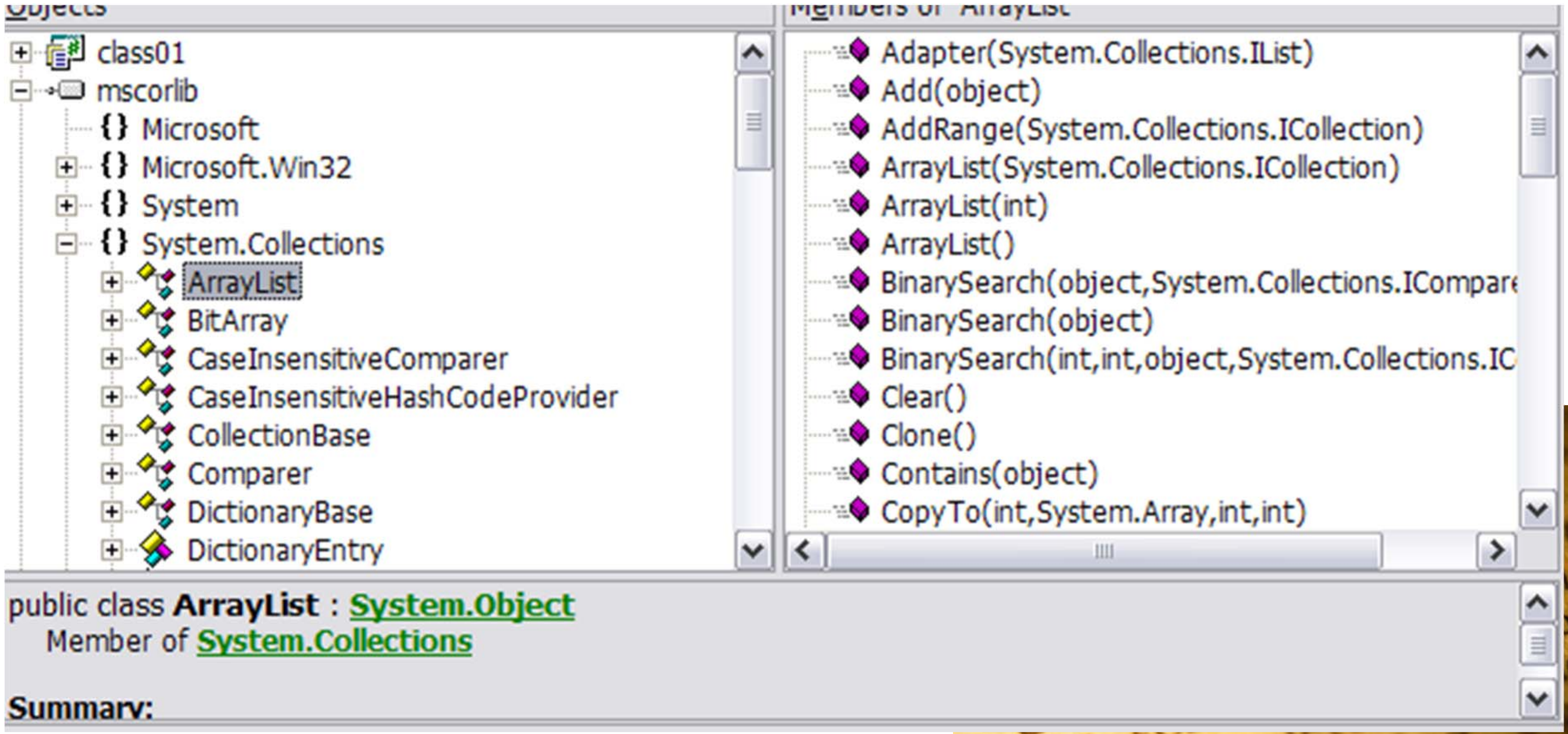
Misc

(Name)	ArrayExample02
Access	internal
FullName	ConsoleApplication2.ArrayExample02
IsAbstract	False

Misc

Debug

Object Browser



Objects

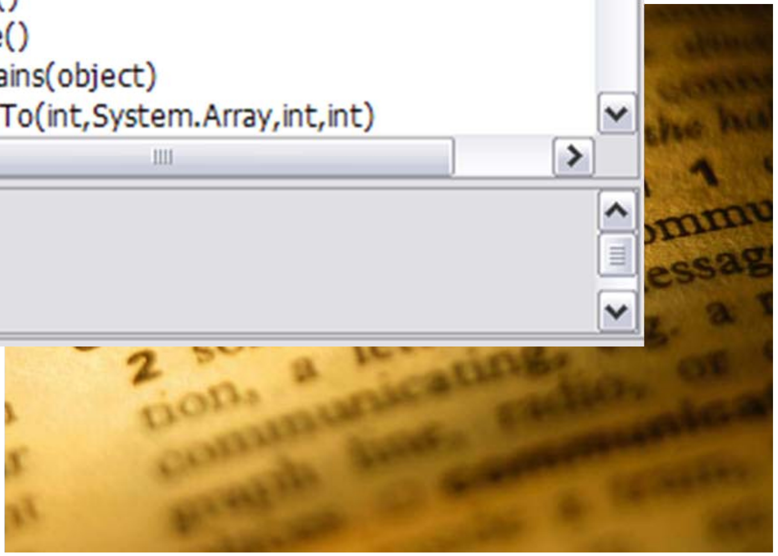
```
class01
mscorlib
  {} Microsoft
  {} Microsoft.Win32
  {} System
  {} System.Collections
     ArrayList
     BitArray
     CaseInsensitiveComparer
     CaseInsensitiveHashCodeProvider
     CollectionBase
     Comparer
     DictionaryBase
     DictionaryEntry
```

Members of 'ArrayList'

```
Adapter(System.Collections.IList)
Add(object)
AddRange(System.Collections.ICollection)
ArrayList(System.Collections.ICollection)
ArrayList(int)
ArrayList()
BinarySearch(object,System.Collections.ICompare
BinarySearch(object)
BinarySearch(int,int,object,System.Collections.IC
Clear()
Clone()
Contains(object)
CopyTo(int,System.Array,int,int)
```

public class **ArrayList** : System.Object
Member of System.Collections

Summary:



System.Collections

.NET Languages

What are the languages?

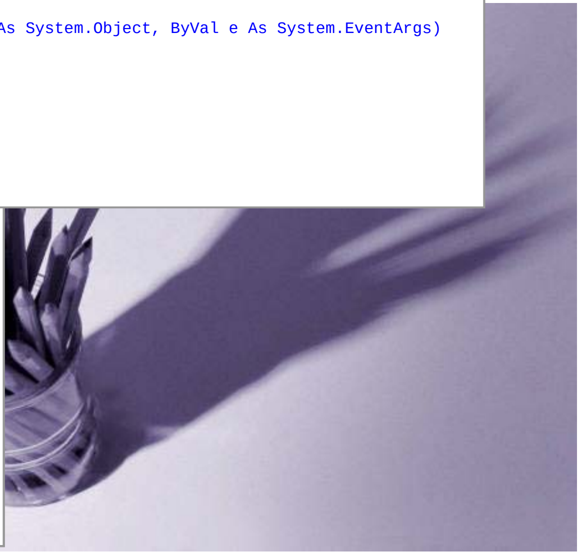
```
' VB.NET Code
Dim j As Integer
      Dim prime As Boolean
      Dim i As Integer

      For i = 1 To 100
          prime = True

          For j = 2 To (i / 2)
              If ((i Mod j) = 0) Then
                  prime = False
          End If
      Next j
          If (prime = True) Then
              TextBox1.Text = TextBox1.Text & "," & Str(i)
          End If
      Next i
```

```
// C# Code
int i, j;
bool prime;

for (i=0;i<100;i++)
{
    prime = true;

    for (j=2;j<=i/2;j++)
    {
        if ((i%j)==0) prime=false;
    }
    if (prime==true) textBox1.Text+=" " +
                                 Convert.ToString(i);
}
```

Example of C# and VB.NET Code

```
Public Class Form1
    Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

    End Sub
    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub
    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    Friend WithEvents TextBox1 As System.Windows.Forms.TextBox
    Friend WithEvents Button1 As System.Windows.Forms.Button
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
        Me.TextBox1 = New System.Windows.Forms.TextBox
        Me.Button1 = New System.Windows.Forms.Button
        Me.SuspendLayout()
        '
        'TextBox1
        '
        Me.TextBox1.Location = New System.Drawing.Point(24, 16)
        Me.TextBox1.Multiline = True
        Me.TextBox1.Name = "TextBox1"
        Me.TextBox1.Size = New System.Drawing.Size(200, 168)
        Me.TextBox1.TabIndex = 0
        Me.TextBox1.Text = ""
        '
        'Button1
        '
        Me.Button1.Location = New System.Drawing.Point(200, 192)
        Me.Button1.Name = "Button1"
        Me.Button1.Size = New System.Drawing.Size(80, 56)
        Me.Button1.TabIndex = 1
        Me.Button1.Text = "E&xit"
        '
        'Form1
        '
        Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
        Me.ClientSize = New System.Drawing.Size(292, 266)
        Me.Controls.Add(Me.Button1)
        Me.Controls.Add(Me.TextBox1)
        Me.Name = "Form1"
        Me.Text = "Form1"
        Me.ResumeLayout(False)

    End Sub

#End Region
```

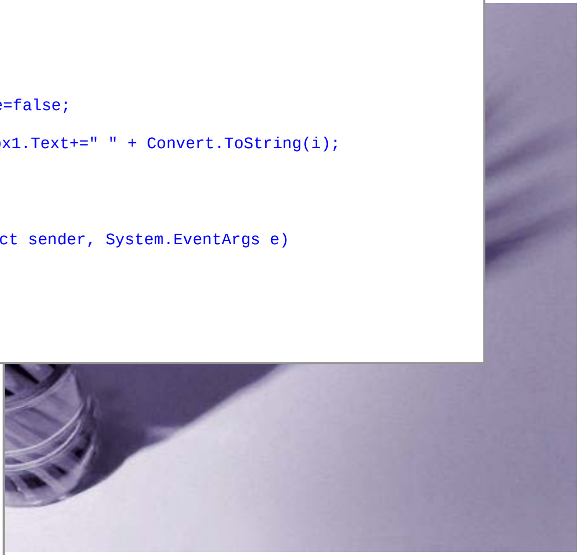
```
 Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox1.TextChanged

    End Sub

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        Dim j As Integer
        Dim prime As Boolean
        Dim i As Integer

        For i = 1 To 100
            prime = True

            For j = 2 To (i / 2)
                If ((i Mod j) = 0) Then
                    prime = False
                End If
            Next j
            If (prime = True) Then
                TextBox1.Text = TextBox1.Text & "," & Str(i)
            End If
        Next i

    End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Application.Exit()
    End Sub
End Class
```



Example VB.NET Code

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

namespace WindowsApplication1
{
        public class Form1 : System.Windows.Forms.Form
        {
                private System.Windows.Forms.TextBox textBox1;
                private System.Windows.Forms.Button button1;
                /// <summary>
                /// Required designer variable.
                /// </summary>
                private System.ComponentModel.Container components = null;

                public Form1()
                {
                        InitializeComponent();

                }

                protected override void Dispose( bool disposing )
                {
                        if( disposing )
                        {
                                if (components != null)
                                {
                                        components.Dispose();
                                }
                        }
                        base.Dispose( disposing );
                }

                #region Windows Form Designer generated code
                private void InitializeComponent()
                {
                        this.textBox1 = new System.Windows.Forms.TextBox();
                        this.button1 = new System.Windows.Forms.Button();
                        this.SuspendLayout();
                        //
                        // textBox1
                        //
                        this.textBox1.Location = new System.Drawing.Point(24, 16);
                        this.textBox1.Multiline = true;
                        this.textBox1.Name = "textBox1";
                        this.textBox1.Size = new System.Drawing.Size(184, 152);
                        this.textBox1.TabIndex = 0;
                        this.textBox1.Text = "";
                        //
                        // button1
                        //
                        this.button1.Location = new System.Drawing.Point(200, 208);
                        this.button1.Name = "button1";
                        this.button1.Size = new System.Drawing.Size(72, 48);
                        this.button1.TabIndex = 1;
                        this.button1.Text = "E&xit";
                        this.button1.Click += new System.EventHandler(this.button1_Click);
                        //
```

```
                        // Form1
                        //
                        this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
                        this.ClientSize = new System.Drawing.Size(292, 266);
                        this.Controls.Add(this.button1);
                        this.Controls.Add(this.textBox1);
                        this.Name = "Form1";
                        this.Text = "Form1";
                        this.Load += new System.EventHandler(this.Form1_Load);
                        this.ResumeLayout(false);

                }
                #endregion
                /// </summary>
                [STAThread]
                static void Main()
                {
                        Application.Run(new Form1());
                }

                private void Form1_Load(object sender, System.EventArgs e)
                {
                        int i, j;
                        bool prime;

                        for (i=0;i<100;i++)
                        {
                                prime = true;
                                for (j=2;j<=i/2;j++)
                                {
                                        if ((i%j)==0) prime=false;
                                }
                                if (prime==true) textBox1.Text+=" " + Convert.ToString(i);
                        }

                }

                private void button1_Click(object sender, System.EventArgs e)
                {
                        Application.Exit();
                }
        }
}
```



Example C# code

Benefits

Why C#?

C Language

```
         <stdio.h>
#include  <math.h>
int       main(void)
{
float  a,b,c,real1,real2,imag;
    puts("Program to determine roots of a quadratic equation");
    printf("Enter a,b and c >>>");
    scanf("%f %f %f",&a,&b,&c);
    printf("Equation is %.2fx*x + %.2fx + %.2f\n",a,b,c);
    if ((b*b)==(4*a*c))
    {  real1=-b/(2*a);
       printf("Root is %.2f\n",real1);
    } else if ((b*b)>(4*a*c))
    {
       real1=(-b+sqrt( (b*b)-4*a*c )) /(2*a);
       real2=(-b-sqrt( (b*b)-4*a*c )) /(2*a);
       printf("Roots are %.2f, %.2f\n",real1,real2);
    } else

    {
       real1=-b/(2*a);
       imag=sqrt(4*a*c-b*b)/(2*a);
       printf("Roots are %.2f +/- j%.2f\n",real1,imag)
    }
    return(0);
}
```

Advantages:
-Minimal language.
-Standardized.
-Flexible.

Disadvantages:
-Weak checking for errors.
-Focused on procedures rather than data.
-Lack of support for graphics (such as Windows).

C Language

C++ Language

← Object-orientation added

```
#include  <iostrea
class circuit
{
private:
   float rtemp;
public:
  float parallel(float r1, float r2)
  {
    return((r1*r2)/(r1+r2));
  }
  float series(float r1, float r2)
  {
    return(r1+r2);
  }
};
int main(void)
{
circuit c1;
float  res;
  res=c1.series(2000,1000);
  cout << "Series resistance is " << res << "ohms\n";
  res=c1.parallel(1000,1000);
  cout << "Parallel resistance is " << res << "ohms\n"
  return(0);
}
```

Advantages:
-Standardized.
-Flexible.
-Object-oriented.
-Improved error checking.
-Improved Windows support

Disadvantages:
-Still a hybrid language (C and/or C++).
-Still too generic.
-Lack of integration with other languages.

C Programming … to …

C Language

C++ Language

C# Language

Windows/
WWW/
Java ideas

```
using System;

namespace ConsoleApplication2
{
  public class Circuit
  {
   public double Parallel(double r1, double r2)
   {
     return((r1*r2)/(r1+r2));
   }
   public double Series(double r1, double r2)
   {
     return(r1+r2);
   }
  }
  class Class1
  {
   static void Main(string[] args)
   {
    double v1=100,v2=100;
    double res;
       Circuit cir = new Circuit();
       res=cir.Parallel(v1,v2);
       System.Console.WriteLine("Parallel resistance i
       res=cir.Series(100,100);
       System.Console.WriteLine("Series resistance is
       System.Console.ReadLine();
    }
}
```

Advantages:
-Fully object-oriented.
-Robust.
-Integrated with Windows.
-Cross-platform.
-Support for mobility.
-Strong integration with

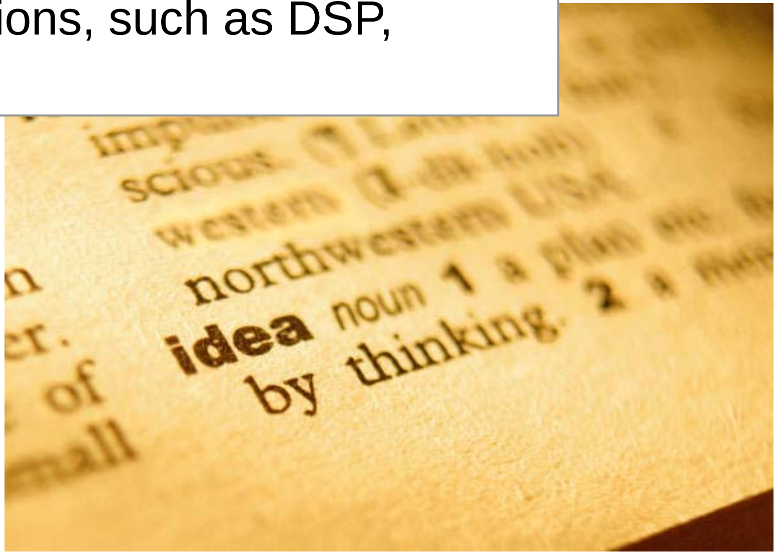
Disadvantages:
-Massive programming environment.

C

The gap between C and VB is now closed as both provide an excellent environment for software development.

VB.NET is aimed at **Microsoft Office** and **WWW-based** Applications, as it integrates well with **VBA** and **ASP**. VB has traditionally supported unstructured code, but this has now changed.

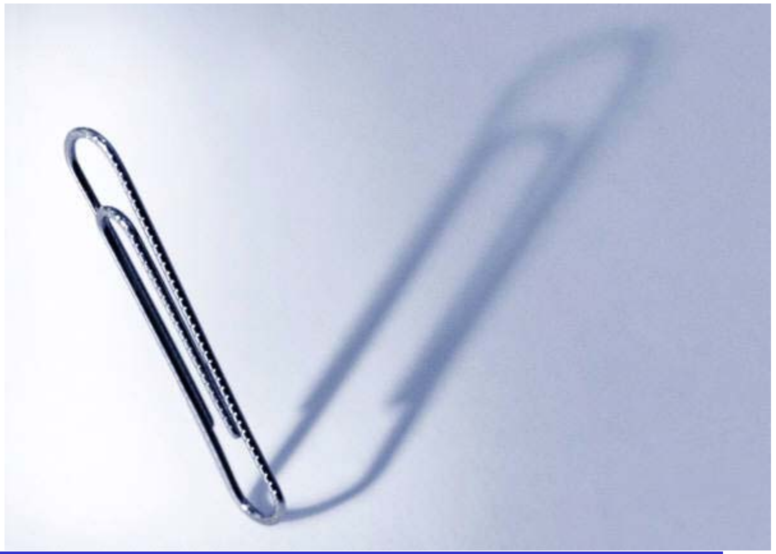
C# is aimed at engineering applications, and allows for more flexibility, such as using pointers. There is also a great amount of code developed for many different applications, such as DSP, interfacing, and so on.



Why C#?

Elements of a C# Program

What goes where?



Why?

```
using System;
namespace ConsoleApplication2
{
      public class Complex
      {
            public double real;
            public double imag;
            public int val { set {} get {} };
            public double mag()
            {
                  return (Math.Sqrt(real*real+imag*imag));
            }
            public double angle()
            {
                  return (Math.Atan(imag/real)*180/Math.PI);
            }
      }
      class Class1
      {
            static void Main(string[] args)
            {
                  Complex r = new Complex();
                  string str;
                  double mag,angle;

                  System.Console.Write("Enter real value >> ");
                  str=System.Console.ReadLine();
                  r.real = Convert.ToInt32(str);
                  System.Console.Write("Enter imag value >> ");
                  str=System.Console.ReadLine();
                  r.imag = Convert.ToInt32(str);
                  mag=r.mag();
                  angle=r.angle();
                  System.Console.WriteLine("Mag is {0} and angle is {1}",mag,angle);
                  System.Console.ReadLine();
            }
      }
}
```

using. Imports types defined in other namespaces.

namespace. Defines a unique name for the objects. In this case the objects would have the name of:
ConsoleApplications2.Complex()
ConsoleApplicaitons2.Class1()

Main(). This is the entry point into the program, and defines the start and end of the program. It must be declared inside a class, and must be static.

C# Program Outline

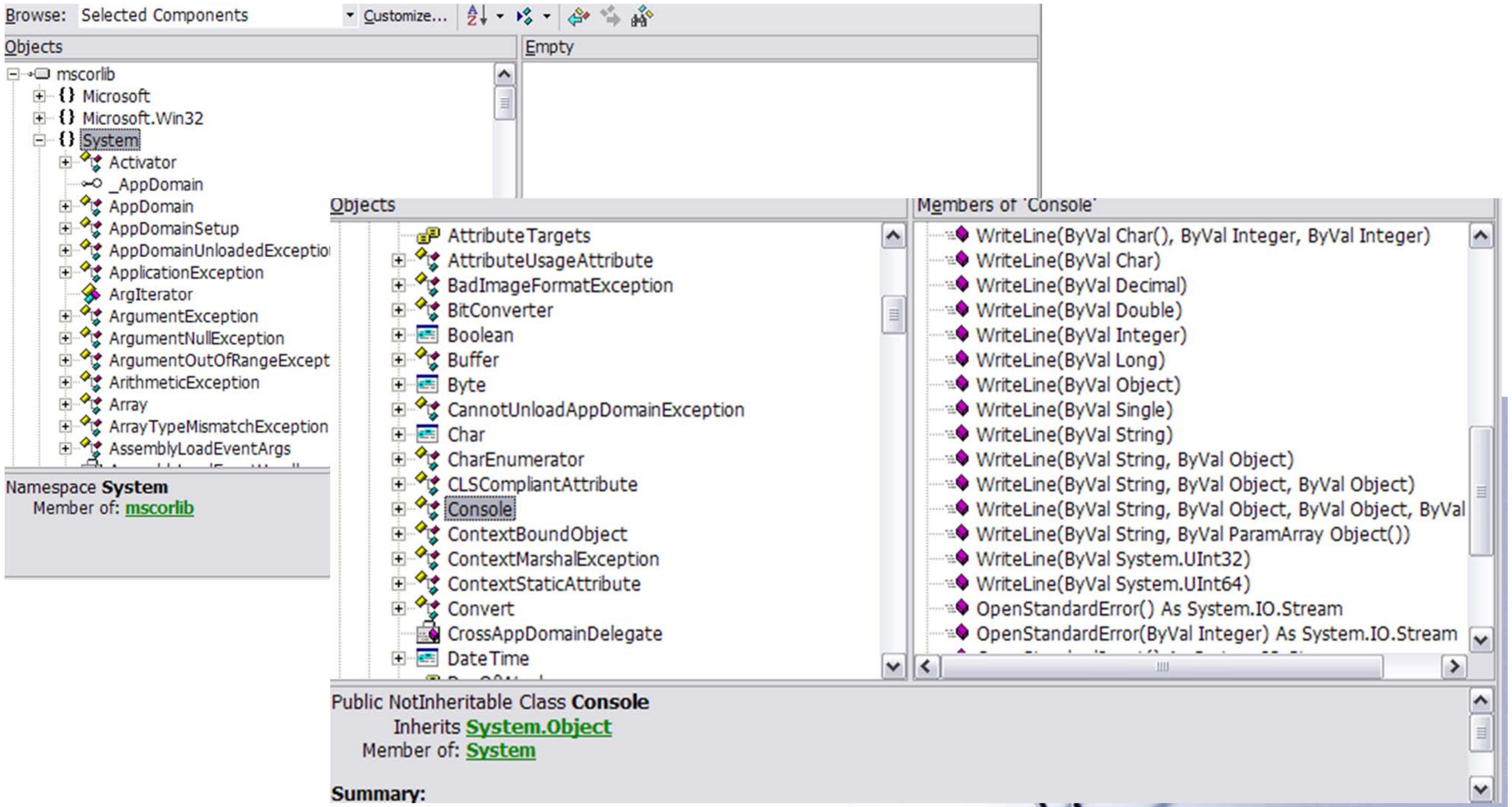
```
using System;
namespace ConsoleApplication2
{
      public class Complex
      {
            public double real;
            public double imag;
            public int val { set {} get {} };
            public double mag()
            {
                  return (Math.Sqrt(real*real+imag*imag));
            }
            public double angle()
            {
                  return (Math.Atan(imag/real)*180/Math.PI);
            }
      }
      class Class1
      {
            static void Main(string[] args)
            {
                  Complex r = new ConsoleApplication2.Complex();
                  string str;
                  double mag,angle;

                  System.Console.Write("Enter real value >> ");
                  str=System.Console.ReadLine();
                  r.real = Convert.ToInt32(str);
                  System.Console.Write("Enter imag value >> ");
                  str=System.Console.ReadLine();
                  r.imag = Convert.ToInt32(str);
                  mag=r.mag();
                  angle=r.angle();
                  System.Console.WriteLine("Mag is {0} and angle is {1}",mag,angle);
                  System.Console.ReadLine();
            }
      }
}
```

namespace. Defines a unique name for the objects. In this case the objects would have the name of:
ConsoleApplications2.Complex()
ConsoleApplicaitons2.Class1()

C# Program Outline

```
System.Console.Write("Enter real value >>
```

using. Imports types defined in other namespaces.

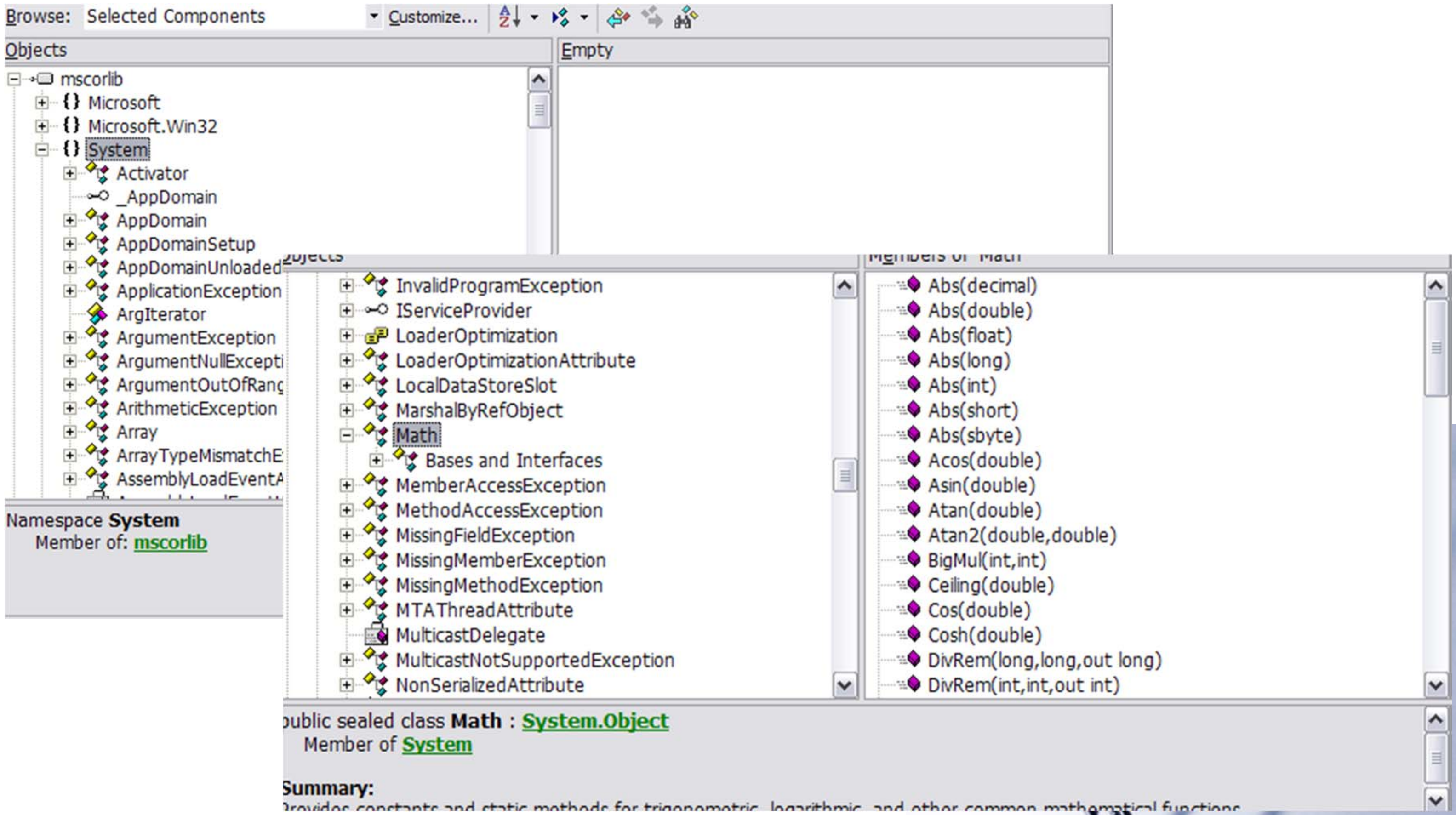


```
Browse: Selected Components   Customize...

Objects
⊟ mscorlib
  ⊞ {} Microsoft
  ⊞ {} Microsoft.Win32
  ⊟ {} System
     ⊞ Activator
        _AppDomain
     ⊞ AppDomain
     ⊞ AppDomainSetup
     ⊞ AppDomainUnloadedExceptio
     ⊞ ApplicationException
        ArgIterator
     ⊞ ArgumentException
     ⊞ ArgumentNullException
     ⊞ ArgumentOutOfRangeExcept
     ⊞ ArithmeticException
     ⊞ Array
     ⊞ ArrayTypeMismatchException
     ⊞ AssemblyLoadEventArgs

Namespace System
  Member of: mscorlib

Empty

Objects
     AttributeTargets
  ⊞ AttributeUsageAttribute
  ⊞ BadImageFormatException
  ⊞ BitConverter
  ⊞ Boolean
  ⊞ Buffer
  ⊞ Byte
  ⊞ CannotUnloadAppDomainException
  ⊞ Char
  ⊞ CharEnumerator
  ⊞ CLSCompliantAttribute
  ⊞ Console
  ⊞ ContextBoundObject
  ⊞ ContextMarshalException
  ⊞ ContextStaticAttribute
  ⊞ Convert
     CrossAppDomainDelegate
  ⊞ DateTime

Members of 'Console'
  WriteLine(ByVal Char(), ByVal Integer, ByVal Integer)
  WriteLine(ByVal Char)
  WriteLine(ByVal Decimal)
  WriteLine(ByVal Double)
  WriteLine(ByVal Integer)
  WriteLine(ByVal Long)
  WriteLine(ByVal Object)
  WriteLine(ByVal Single)
  WriteLine(ByVal String)
  WriteLine(ByVal String, ByVal Object)
  WriteLine(ByVal String, ByVal Object, ByVal Object)
  WriteLine(ByVal String, ByVal Object, ByVal Object, ByVal
  WriteLine(ByVal String, ByVal ParamArray Object())
  WriteLine(ByVal System.UInt32)
  WriteLine(ByVal System.UInt64)
  OpenStandardError() As System.IO.Stream
  OpenStandardError(ByVal Integer) As System.IO.Stream

Public NotInheritable Class Console
     Inherits System.Object
  Member of: System

Summary:
```

System.Console.

```
Math.Sqrt(real*real+imag*imag)
```

using. Imports types defined in other namespaces.



mscorlib
Microsoft
Microsoft.Win32
System
Activator
_AppDomain
AppDomain
AppDomainSetup
AppDomainUnloaded
ApplicationException
ArgIterator
ArgumentException
ArgumentNullExcepti
ArgumentOutOfRang
ArithmeticException
Array
ArrayTypeMismatchE
AssemblyLoadEventA

Namespace **System**
Member of: mscorlib

InvalidProgramException
IServiceProvider
LoaderOptimization
LoaderOptimizationAttribute
LocalDataStoreSlot
MarshalByRefObject
Math
Bases and Interfaces
MemberAccessException
MethodAccessException
MissingFieldException
MissingMemberException
MissingMethodException
MTAThreadAttribute
MulticastDelegate
MulticastNotSupportedException
NonSerializedAttribute

Abs(decimal)
Abs(double)
Abs(float)
Abs(long)
Abs(int)
Abs(short)
Abs(sbyte)
Acos(double)
Asin(double)
Atan(double)
Atan2(double,double)
BigMul(int,int)
Ceiling(double)
Cos(double)
Cosh(double)
DivRem(long,long,out long)
DivRem(int,int,out int)

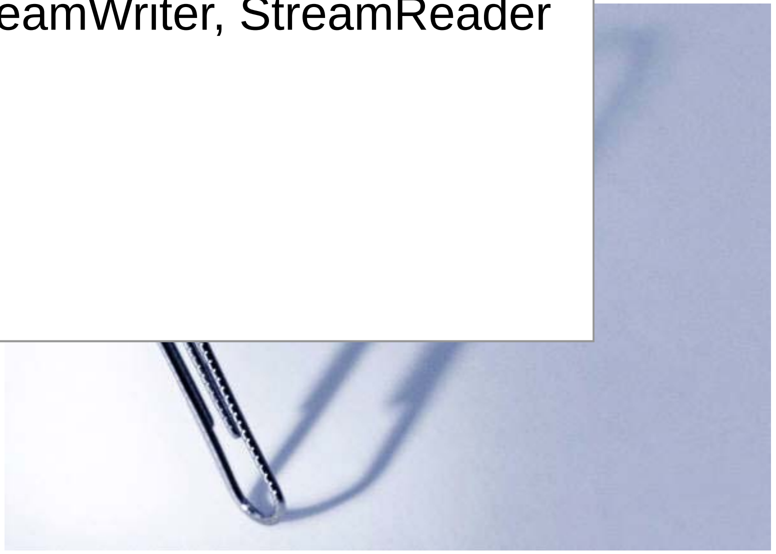
public sealed class **Math** : System.Object
Member of System

Summary:
Provides constants and static methods for trigonometric, logarithmic, and other common mathematical functions.

System.Math.

System:
Array, Boolean, Byte, Char, Convert, DateTime, Double, Enum, Int16, Int32, Int 64, Math, Random, String, Void
System.Collections:
ArrayList, BitArray, Hashtable, Queue, Stack.
System.IO:
BinaryReader, BinaryWriter, File, Stream, StreamWriter, StreamReader

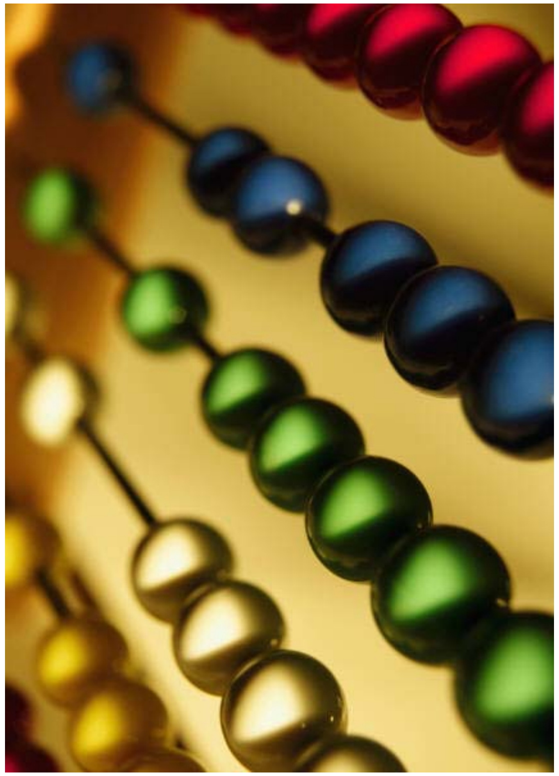
```
uses System;
uses System.Collections;
uses System.IO
```



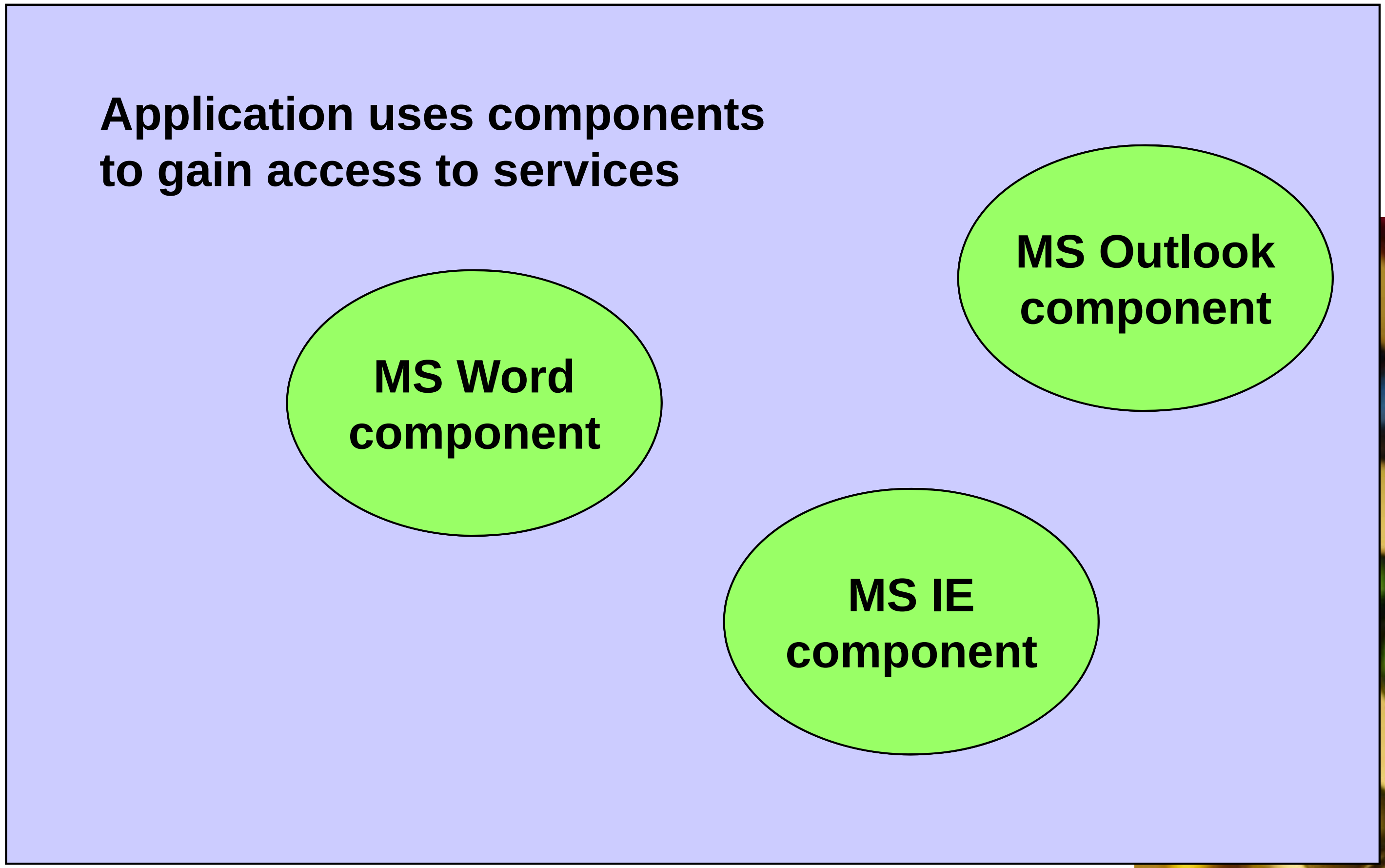
Typical namespaces

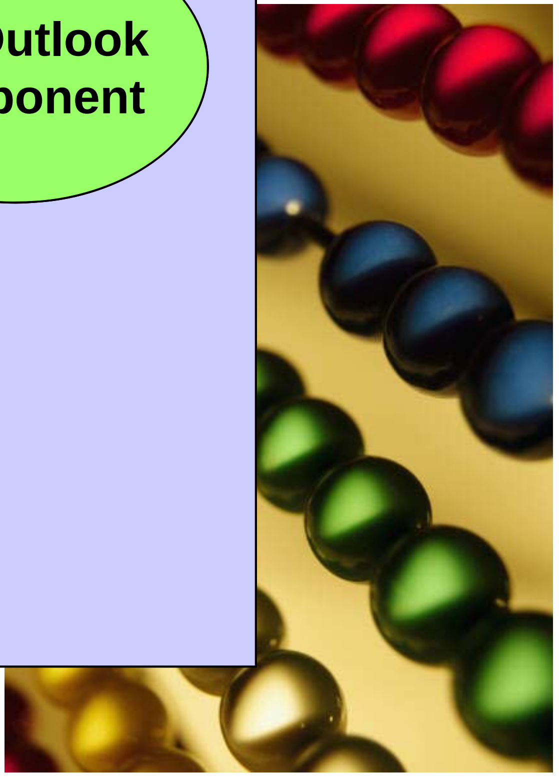
.NET Components

What are components?



Why?

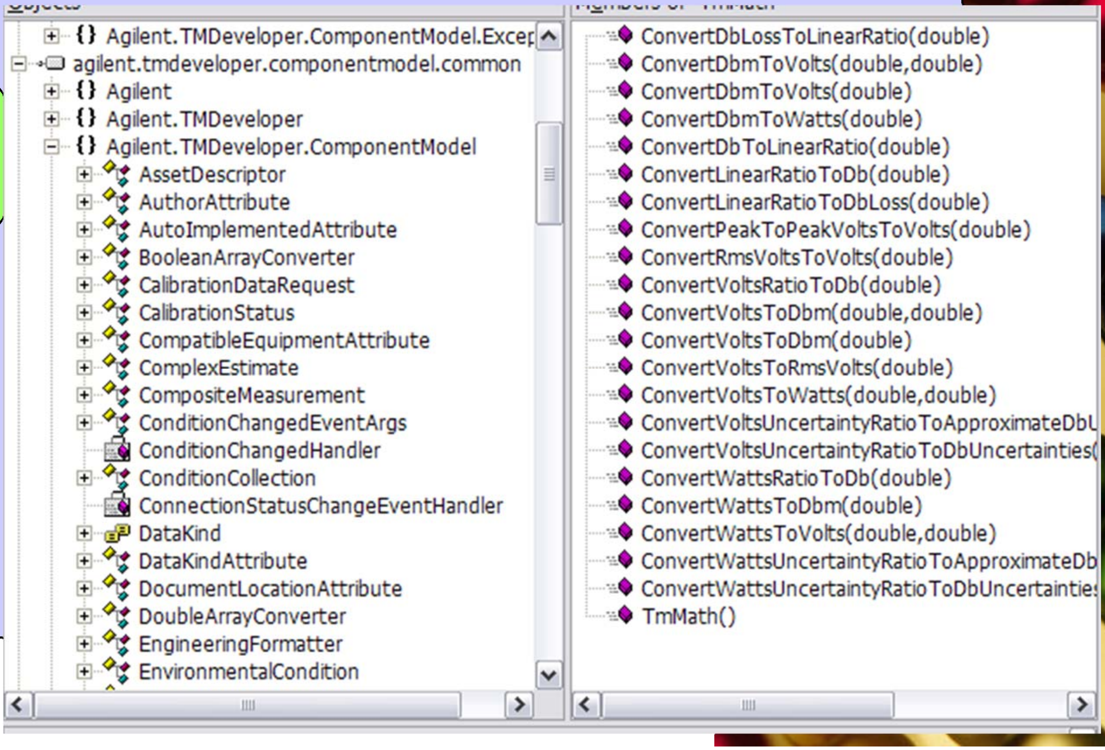

Application uses components
to gain access to services
MS Word
component
MS Outlook
component
MS IE
component



Components

Application uses components to gain access to services

TM Developer component

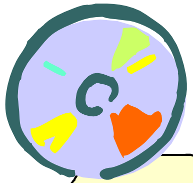

Agilent.TMDeveloper.ComponentModel.Excep
agilent.tmdeveloper.componentmodel.common
Agilent
Agilent.TMDeveloper
Agilent.TMDeveloper.ComponentModel
AssetDescriptor
AuthorAttribute
AutoImplementedAttribute
BooleanArrayConverter
CalibrationDataRequest
CalibrationStatus
CompatibleEquipmentAttribute
ComplexEstimate
CompositeMeasurement
ConditionChangedEventArgs
ConditionChangedHandler
ConditionCollection
ConnectionStatusChangeEventHandler
DataKind
DataKindAttribute
DocumentLocationAttribute
DoubleArrayConverter
EngineeringFormatter
EnvironmentalCondition

ConvertDbLossToLinearRatio(double)
ConvertDbmToVolts(double,double)
ConvertDbmToVolts(double)
ConvertDbmToWatts(double)
ConvertDbToLinearRatio(double)
ConvertLinearRatioToDb(double)
ConvertLinearRatioToDbLoss(double)
ConvertPeakToPeakVoltsToVolts(double)
ConvertRmsVoltsToVolts(double)
ConvertVoltsRatioToDb(double)
ConvertVoltsToDbm(double,double)
ConvertVoltsToDbm(double)
ConvertVoltsToRmsVolts(double)
ConvertVoltsToWatts(double,double)
ConvertVoltsUncertaintyRatioToApproximateDbU
ConvertVoltsUncertaintyRatioToDbUncertainties(
ConvertWattsRatioToDb(double)
ConvertWattsToDbm(double)
ConvertWattsToVolts(double,double)
ConvertWattsUncertaintyRatioToApproximateDb
ConvertWattsUncertaintyRatioToDbUncertainties
TmMath()

TM Developer Components

Tutorial Session 2:

Q1.3 and on

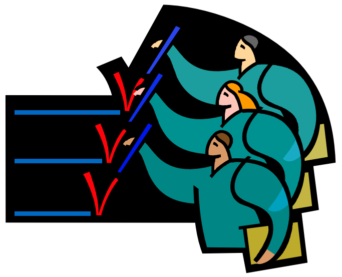


Presentations

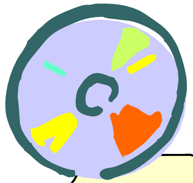
Notes

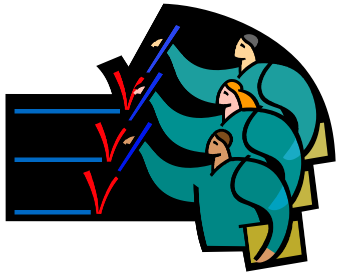
SourceCode

Tutorials



Sample Solutions





Presentations

Notes

SourceCode

Tutorials

```
using System;
namespace solution1_02
{
  class Class1
  {
    static void Main(string[] args)
    {
     string myname;
     System.Console.WriteLine("What is your name >>");
     myname=System.Console.ReadLine();
     System.Console.WriteLine("Your name is  " + myname);
     System.Console.ReadLine();
    }
  }
}
```

```
using System;
namespace solution1_03
{
  class Class1
  {
   static void Main(string[] args)
   {
     double r1,r2,rp,rs;
     string str;
     System.Console.WriteLine("Enter R1 >>");
     str = System.Console.ReadLine();
     r1=System.Convert.ToDouble(str);
     System.Console.WriteLine("Enter R2 >>");
     str = System.Console.ReadLine();
     r2=System.Convert.ToDouble(str);
     rp=(r1*r2)/(r1+r2);
     rs=r1+r2;
     System.Console.WriteLine("Parallel: {0} Ohms,
       Series: {1} Ohms",rp,rs);
   }
  }
}
```

```
using System;
namespace solution1_04
{
 class Class1
 {
   static void Main(string[] args)
   {
     double Pi, Po, Pgain;
     string str;
     System.Console.WriteLine("Enter Pin >>");
     str = System.Console.ReadLine();
     Pi = System.Convert.ToDouble(str);
     System.Console.WriteLine("Enter Pout >>");
     str = System.Console.ReadLine();
     Po = System.Convert.ToDouble(str);
     Pgain = 10 * Math.Log10(Po/Pi);
     System.Console.WriteLine("Gain is {0} dB",Pgain);
   }
 }
}
```

```
using System;
namespace solution1_5
{
  class Class1
  {
    static void Main(string[] args)
    {
      double Vi, Vo, Pgain;
      string str;
      System.Console.WriteLine("Enter Vin >>");
      str = System.Console.ReadLine();
      Vi = System.Convert.ToDouble(str);
      System.Console.WriteLine("Enter Vout >>");
      str = System.Console.ReadLine();
      Vo = System.Convert.ToDouble(str);
      Pgain = 20 * Math.Log10(Vo/Vi);
      System.Console.WriteLine("Gain is {0} dB",Pgain);
    }
  }
}
```

```
using System;
namespace ConsoleApplication1
{
        class Class1
        {
                static void Main(string[] args)
                {
                        WriteLine("This is my first program");
                        ReadLine();
                }
        }
// Problem is here... no closing bracket
```

```
using System;
namespace ConsoleApplication1
{
 class Class1
 {
   static void Main(string[] args)
   {
    double val1 = 10;
    // result not declared...
    result = Math.Sqrt(10);
    System.Console.WriteLine("Square root of 10 is {0} ",
        result);
   }
 }
}
```

```
// This program has two syntax errors
using System;
namespace ConsoleApplication2
{
 public class Cup
 {
   public string Shape;              public string Colour;
   public string Size;               public int Transparency;
   public string Handle;
   public void DisplayCup()
   {
       // Needs opening quotes...
       System.Console.WriteLine(Colour: {0}, Handle: {1}",
                Colour, Handle)

 }
class Class1
{
  static void Main(string[] args)
  {
      Cup cup = new Cup();
      Cup.Colour = "Red";
      cup.Handle = "Small";
      cup.DisplayCup();
      System.Console.ReadLine();
  }
}
}
```

```
using System;
namespace ConsoleApplication2
{
 public class Instrument
 {
   public string Types;
   public string VoltageRange;
   public string PowerRange;
   public void DisplayInstrument()
   {
       System.Console.WriteLine(
   "Instrument is “ Types, VoltageRange);
   }
 }
```

```
e.WriteLine(“Value is {0} {1} “, val1, val2);
e.WriteLine(“Value is “ + val1 + “Value is “ + val
class Class1
{
  static void Main(string[] args)
  {
      Instrument instrument = new Instrument();
      instrument.Types = "ABC01";
      instrument.VoltageRange = "microVolts";
      instrument.DisplayInstrument();
      instrument.Types = "DEF01";
      instrument.VoltageRange = "milliVolts";
      instrument.DisplayInstrument();
  }
}
```