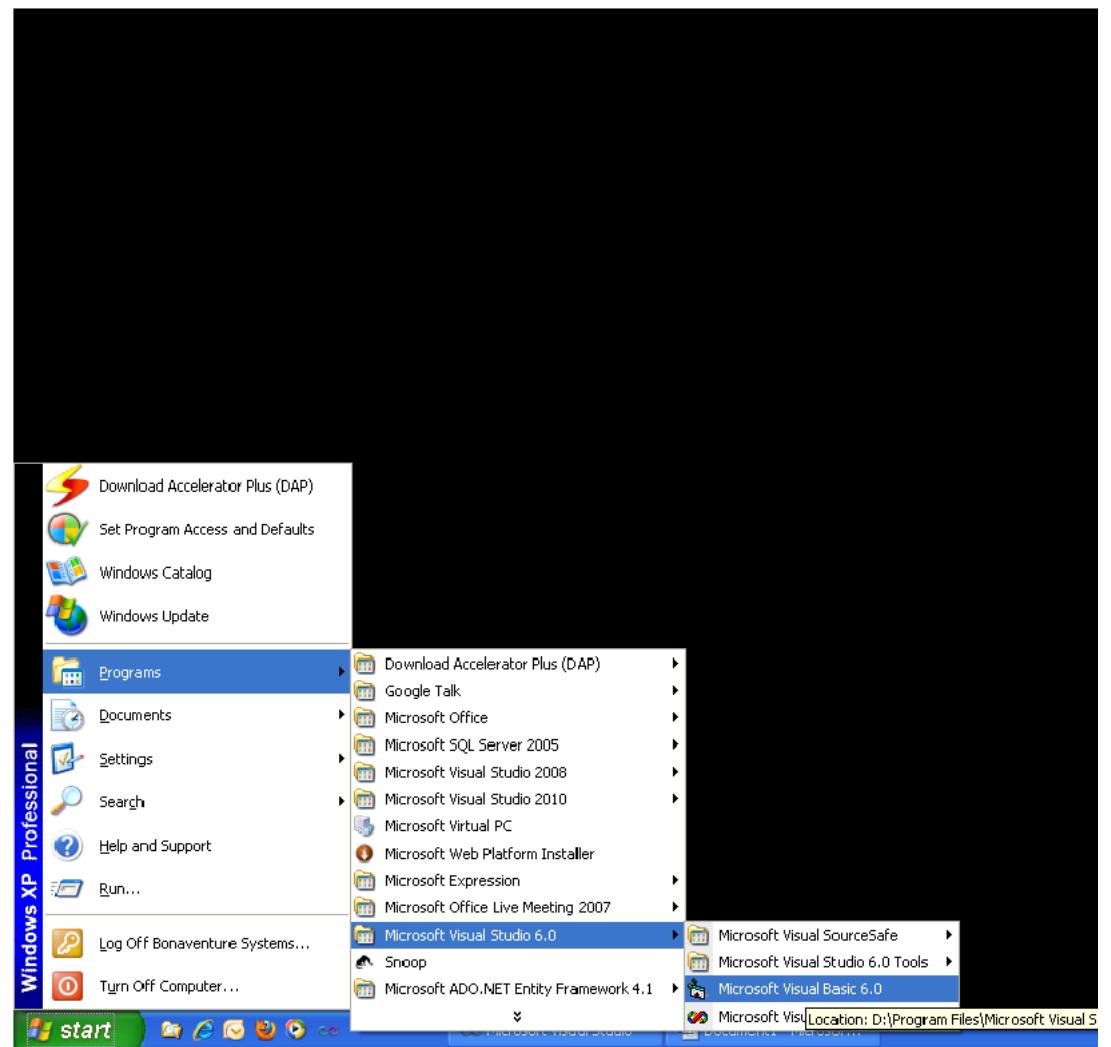
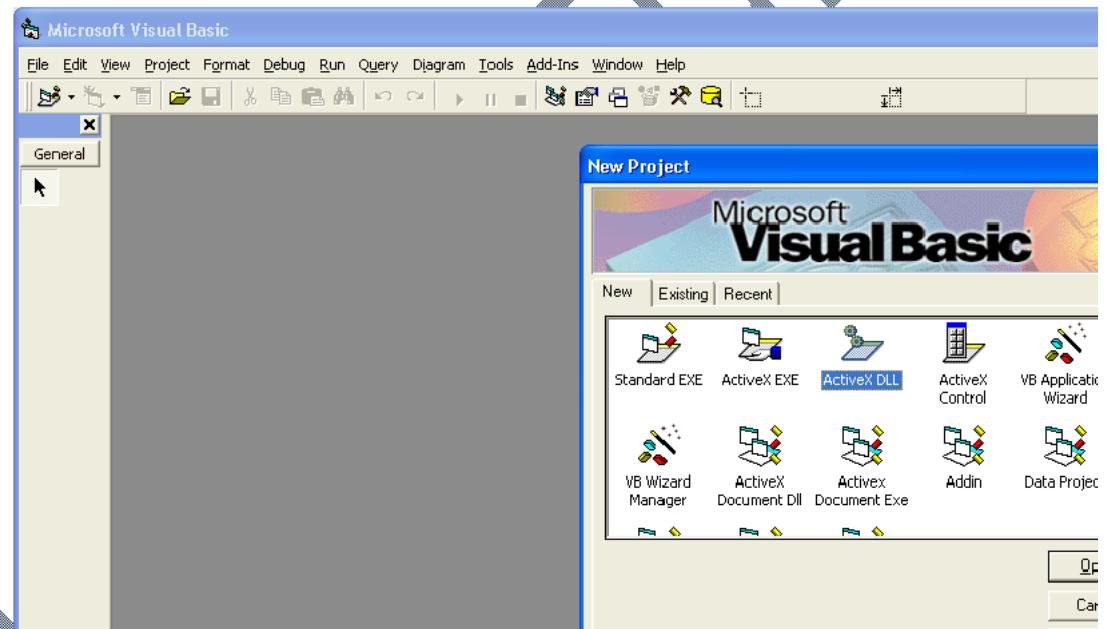




## 1. Start Visual Studio 6.0



2.



Project1 - Microsoft Visual Basic [design] - [Class1 (Code)]

File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help

(General) Mult

```
Public Function Add(x As Integer, y As Integer) As Integer
    Add = x + y
End Function

Public Function Mult(x As Integer, y As Integer) As Integer
    Mult = x * y
End Function
```

4. Microsoft Visual Studio Document1 - Microsoft Visual Studio Project1 - Microsoft Visual Studio

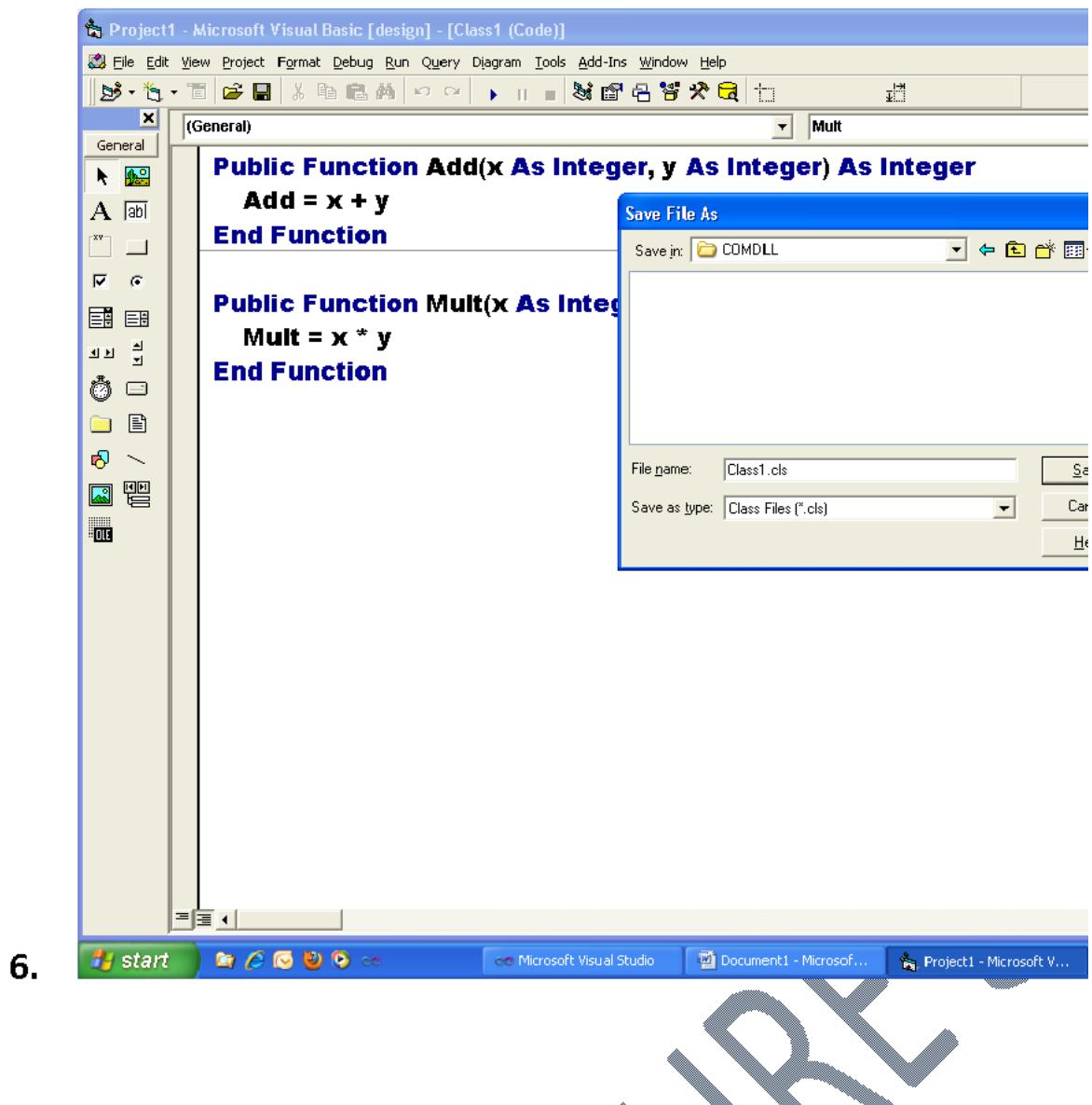
Save File As

Save in: CDAC

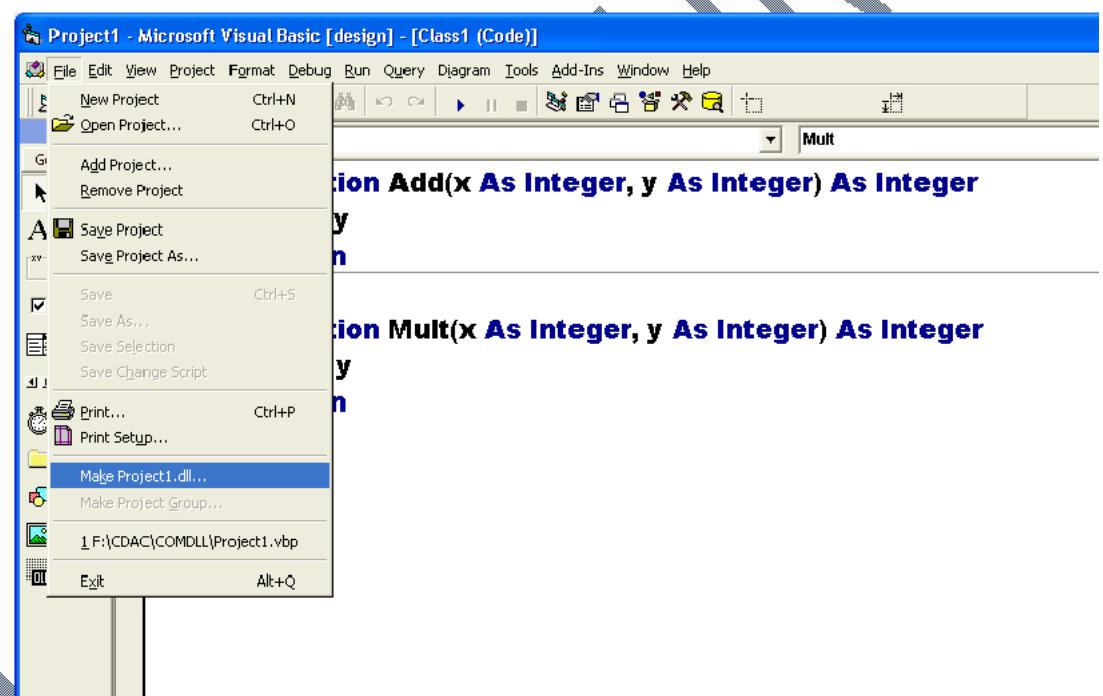
Test\_FileIo  
Test\_FileIO\_BinarySerialization  
Test\_FileIO\_SOAPSerialization  
Test\_FileIO\_XMLSerialization  
COMDLL

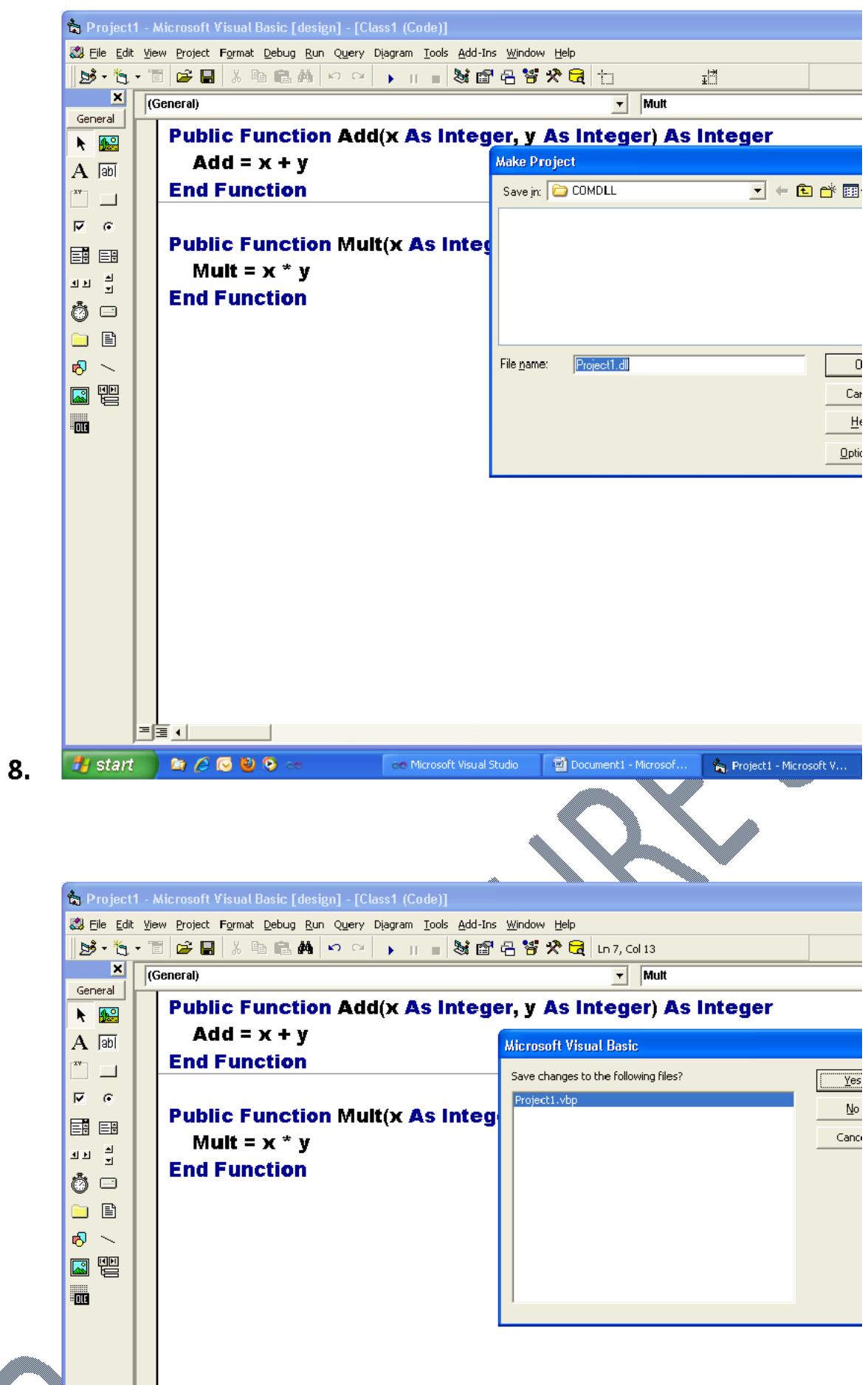
File name: Class1.cls

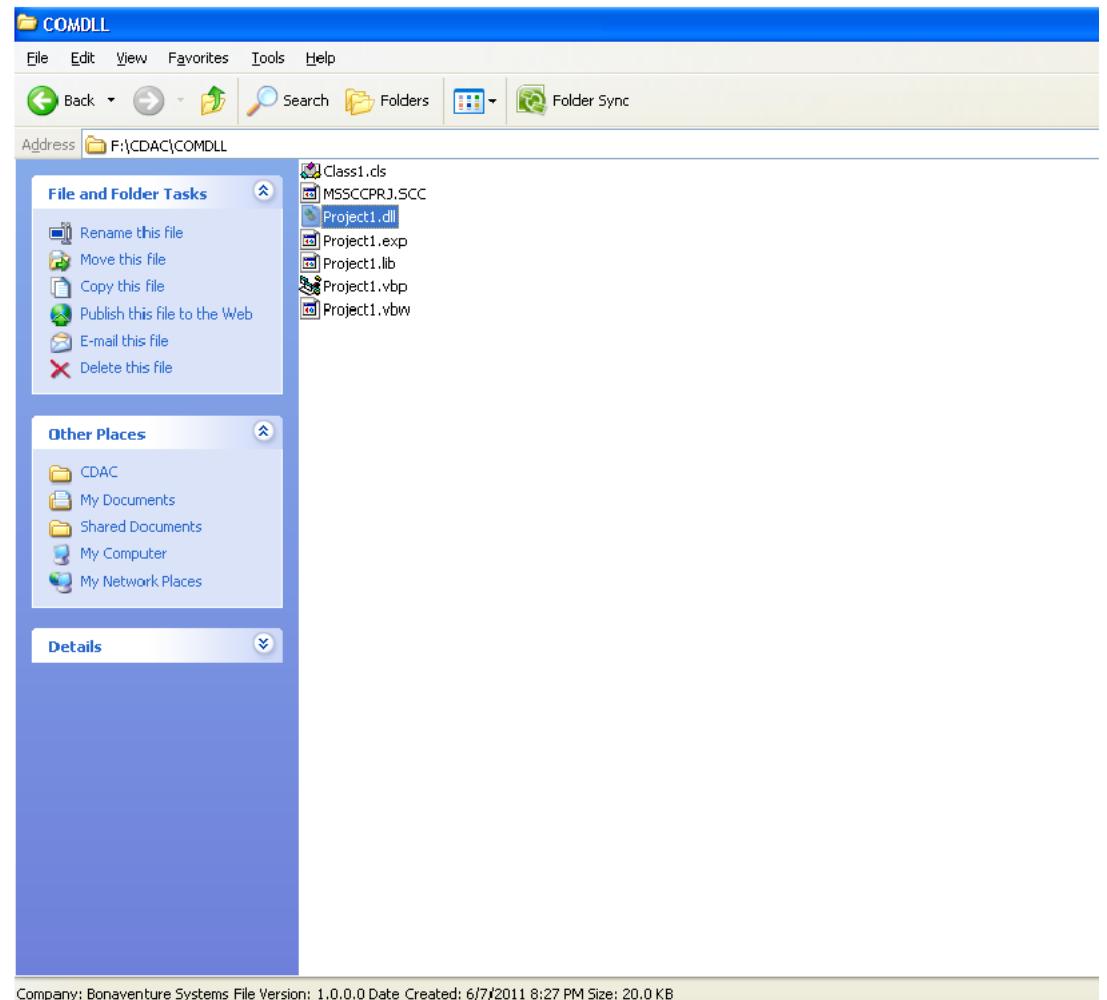
Save as type: Class Files (\*.cls)



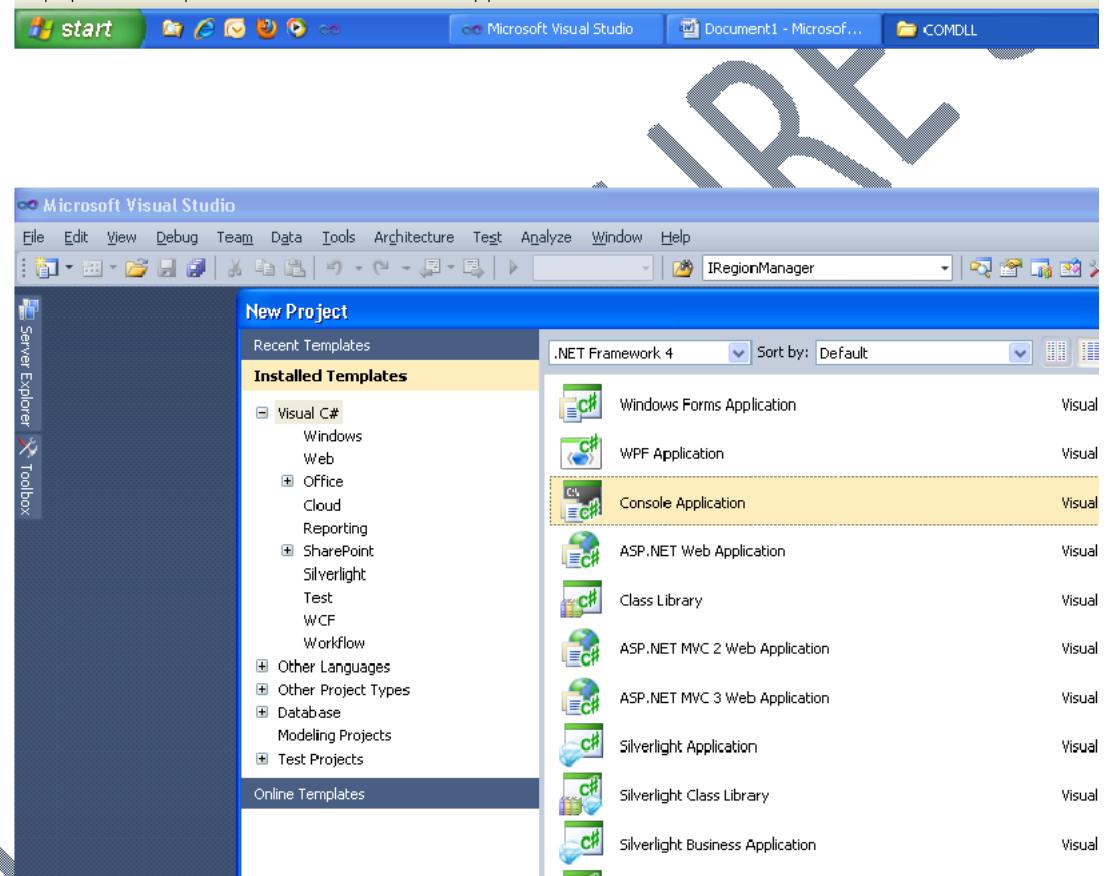
6.







10.





```
File Edit View Refactor Project Build Debug Team Data Tools Architecture Test Analyze Window Help Full Screen
Program.cs* x
COM_CallerApplication.Program
Main(string[] args)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace COM_CallerApplication
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}

100 %
Error List
Ready
```

12.



```
File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help Full Screen
Program.cs* x
COM_CallerApplication.Program
Main(string[] args)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace COM_CallerApplication
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

The screenshot shows the Microsoft Visual Studio IDE. In the center is the code editor for a file named 'Program.cs'. The code defines a class 'Program' with a static void Main method. The code is color-coded, with 'using' statements in blue and class definitions in black. To the right of the code editor is a 'Add Reference' dialog box. The 'COM' tab is selected. A list of available COM components is shown, including 'Project1' at the top. At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace COM_CallerApplication
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

14.

The screenshot shows the Microsoft Visual Studio IDE with a large, semi-transparent 'IPX' watermark overlaid on the interface. The code editor displays the same 'Program.cs' file as the previous screenshot. The code is identical:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace COM_CallerApplication
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

The screenshot shows the 'Program.cs' file in Visual Studio. The code has been partially typed:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace COM_CallerApplication
{
    class Program
    {
        static void Main(string[] args)
        {
            Project1.Class1Class obj = new Project1.Class1Class();
            obj.Add(
        }
    }
}
```

16.

The screenshot shows the 'Program.cs' file in Visual Studio with the Main() method completed:

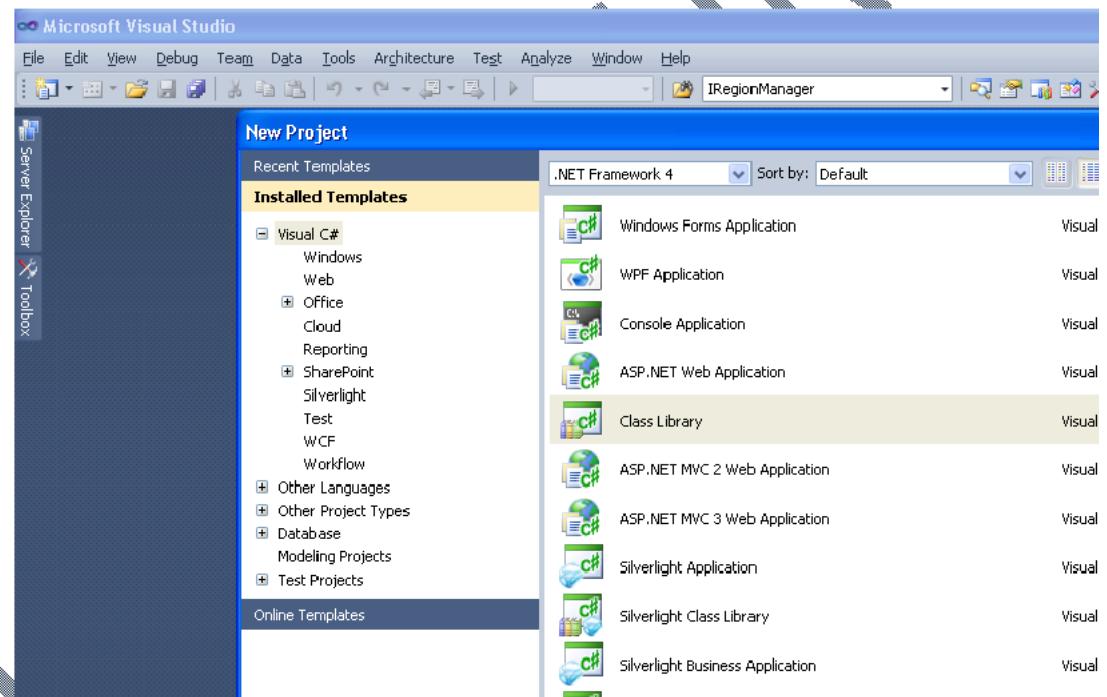
```
namespace COM_CallerApplication
{
    class Program
    {
        static void Main(string[] args)
        {
            Project1.Class1Class obj = new Project1.Class1Class();
            short x, y;
            x = 100;
            y = 200;
            short result = obj.Add(ref x, ref y);
            Console.WriteLine(result.ToString());
            Console.ReadLine();
        }
    }
}
```

The screenshot shows the Microsoft Visual Studio IDE. At the top, the menu bar includes File, Edit, View, Project, Build, Debug, Team, Data, Tools, Architecture, Test, Analyze, Window, Help, and Full Screen. A toolbar is visible below the menu. The main area displays a code editor with the file name Program.cs. The code is as follows:

```
    y = 200;
    short result = obj.Add(ref x, ref y);
    Console.WriteLine(result.ToString());
    Console.ReadLine();
}
```

Below the code editor is a status bar showing "100 %". To the right of the code editor is a "Watch 1" window and an "Error List" window. The "Error List" shows 0 Errors, 0 Warnings, and 0 Messages. The taskbar at the bottom of the screen shows icons for Start, Internet Explorer, and other applications, along with the project names "COM\_CallerApplication" and "COM\_Interoperability".

18.



The screenshot shows the Microsoft Visual Studio IDE interface. The title bar reads "Class1.cs" and "DotNETLibrary.CMath". The menu bar includes File, Edit, View, Refactor, Project, Build, Debug, Team, Data, Tools, Architecture, Test, Analyze, Window, Help, and Full Screen. The code editor displays the following C# code:

```
namespace DotNETLibrary
{
    public class CMath
    {
        public int Add(int x, int y)
        {
            return x + y;
        }
        public int Mult(int x, int y)
        {
            return x * y;
        }
    }
}
```

The status bar at the bottom shows "100 %", "Error List", and "Item(s) Saved".

20.

The screenshot shows the Microsoft Visual Studio IDE interface with the title bar "DotNETLibrary - Microsoft Visual Studio". The code editor displays the following C# code in the AssemblyInfo.cs file:

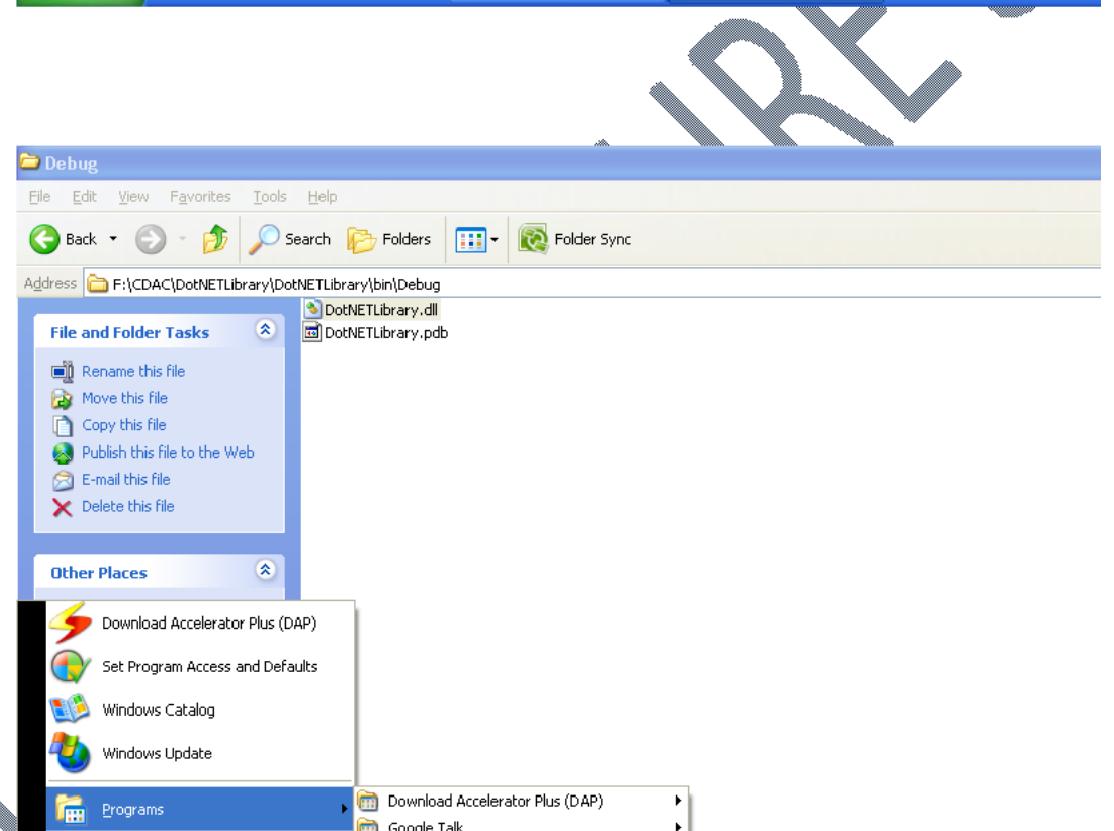
```
using System.Reflection;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;

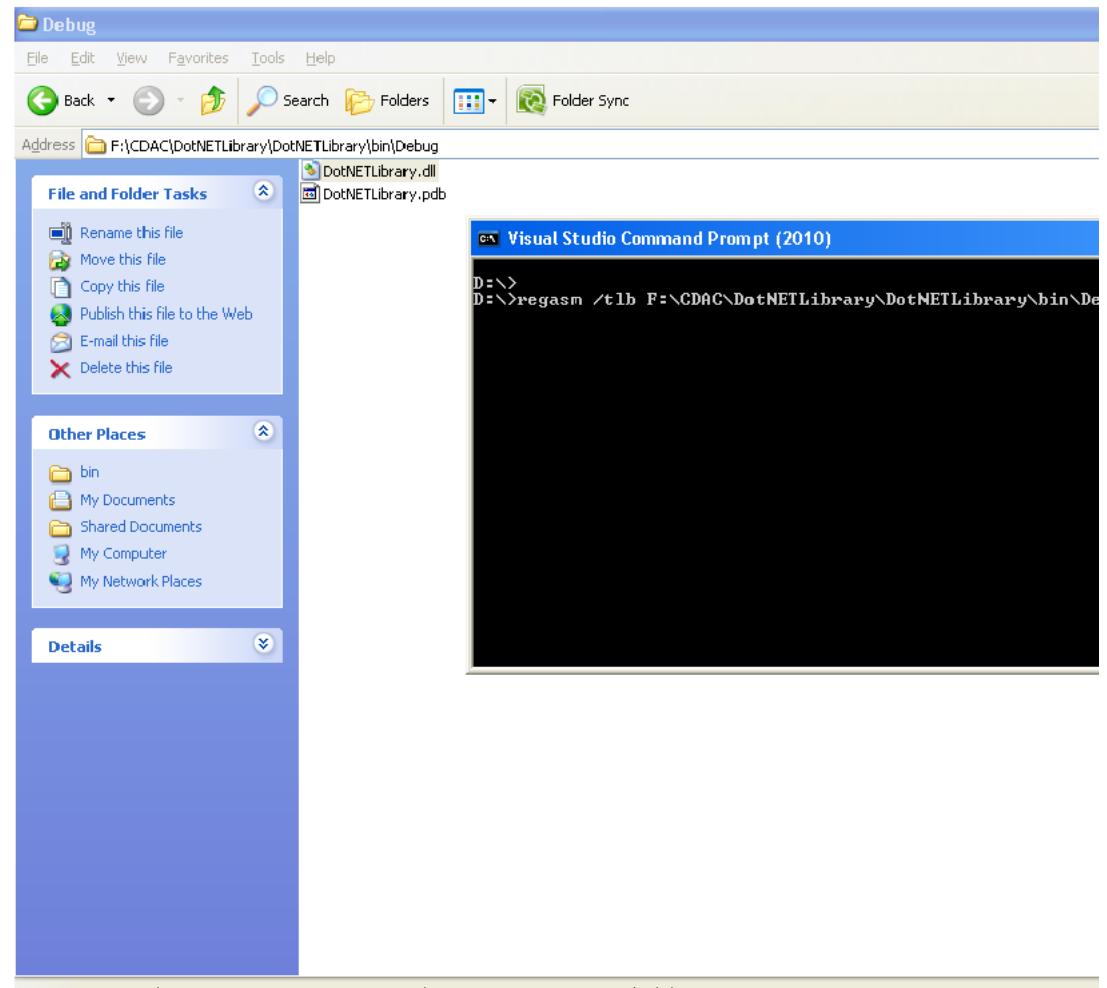
// General Information about an assembly is controlled by
// the set of attributes. Change these attribute values
// to associate an assembly with a specific
// set of attributes.
[assembly: AssemblyTitle("DotNETLibrary")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("Bonaventure Systems")]
```

The screenshot shows the Microsoft Visual Studio interface with the title bar "DotNETLibrary - Microsoft Visual Studio". The menu bar includes File, Edit, View, Refactor, Project, Build, Debug, Team, Data, Tools, Architecture, Test, Analyze, Window, Help. The toolbar has various icons for file operations. The main window shows two tabs: "AssemblyInfo.cs\*" and "Class1.cs". The code editor contains the following C# code:

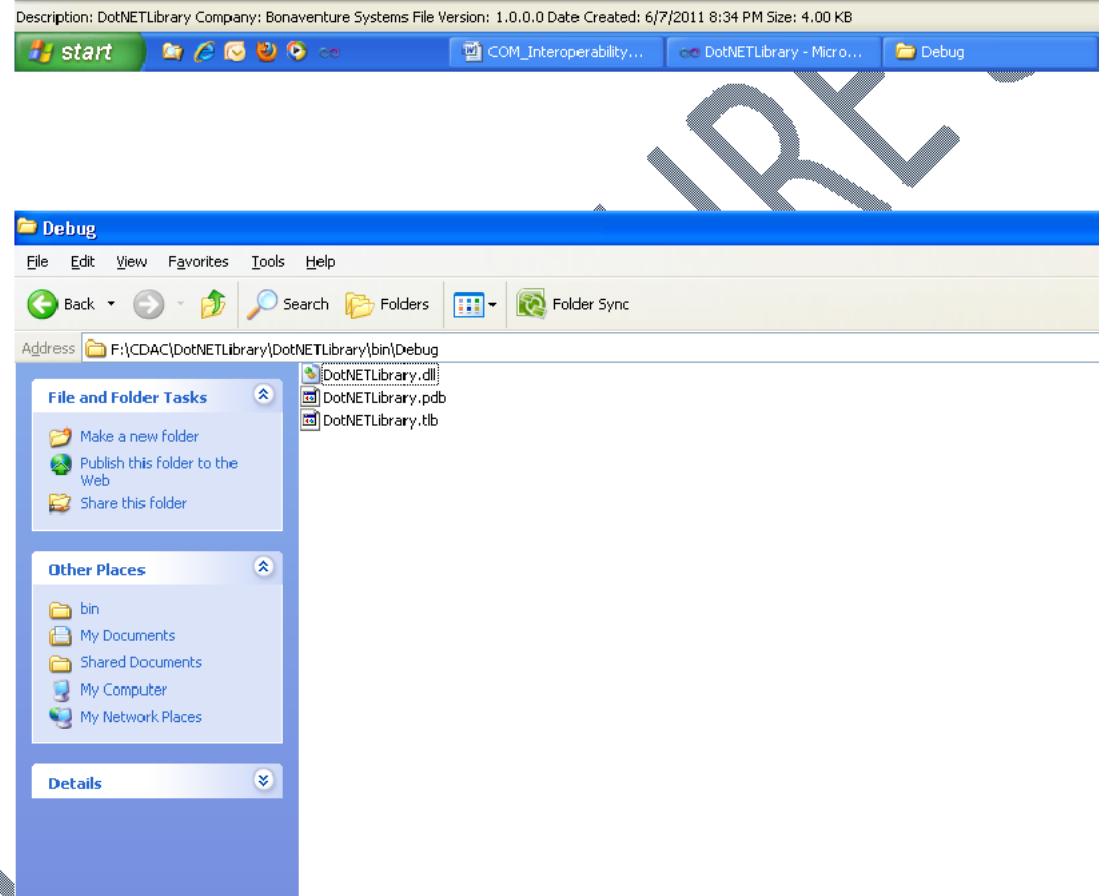
```
// to COM components. If you need to access a type
// COM, set the ComVisible attribute to true on the
// [assembly: ComVisible(true)]
// The following GUID is for the ID of the typelib if
// COM
[assembly: Guid("c2d7280e-26c4-4984-92c6-4723")]
// Version information for an assembly consists of
// Major Version
// Minor Version
// Build Number
// Revision
```

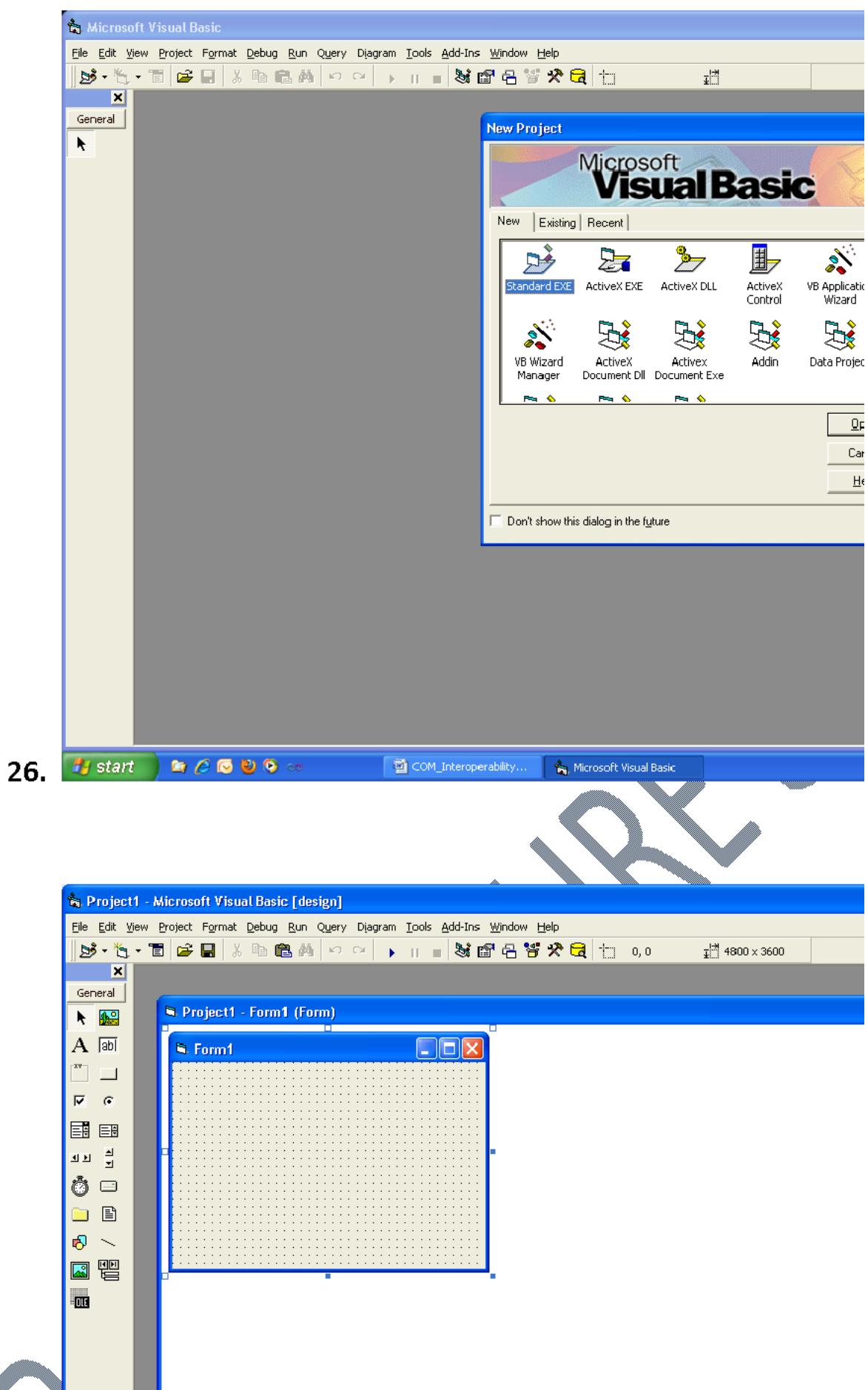
22.

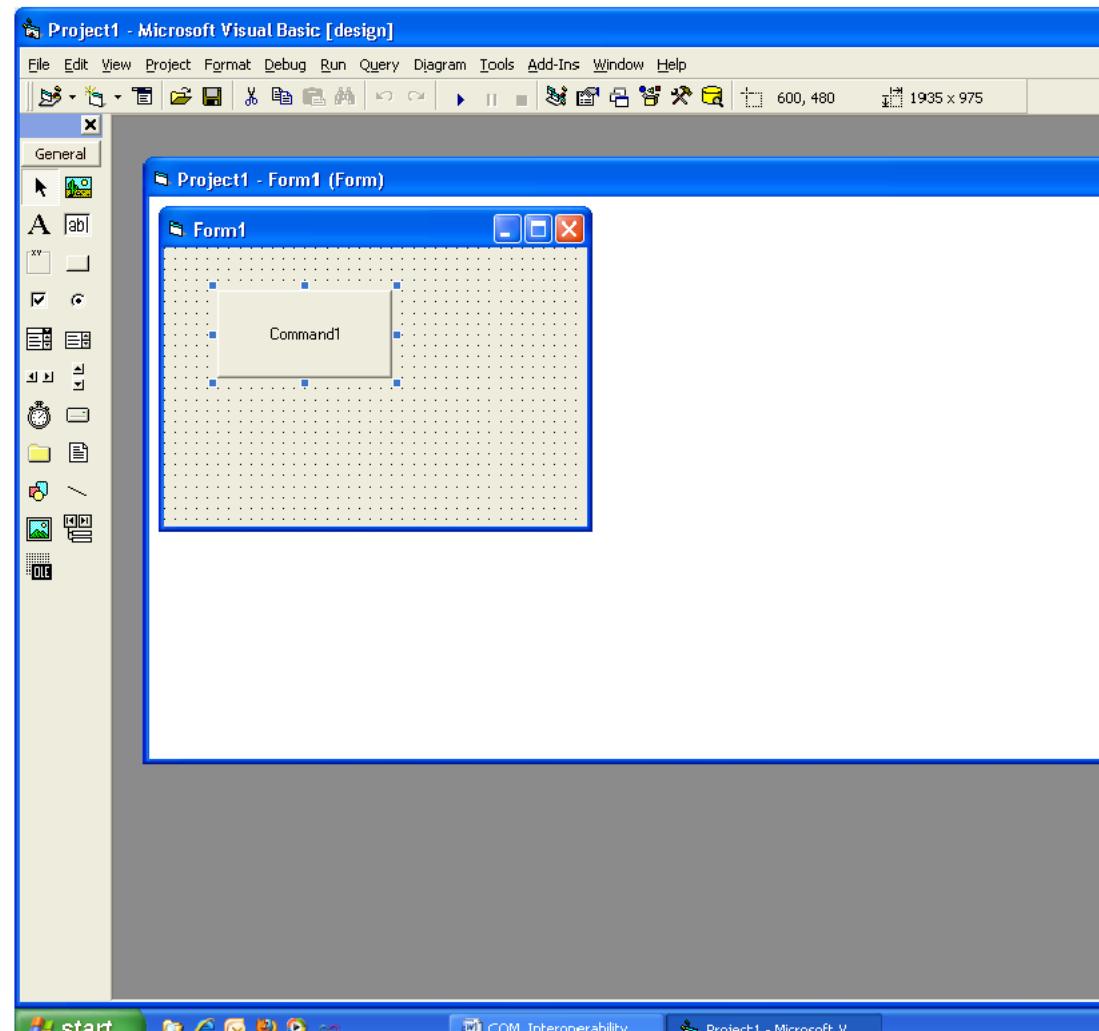




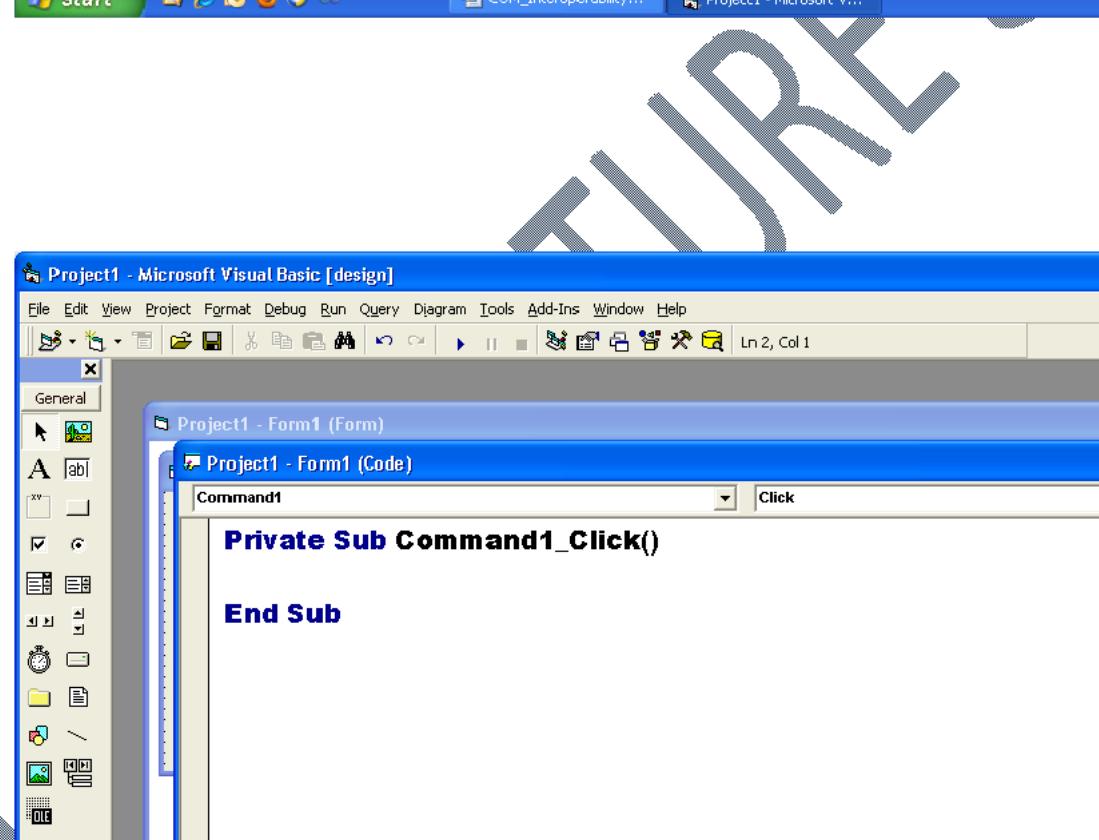
24.

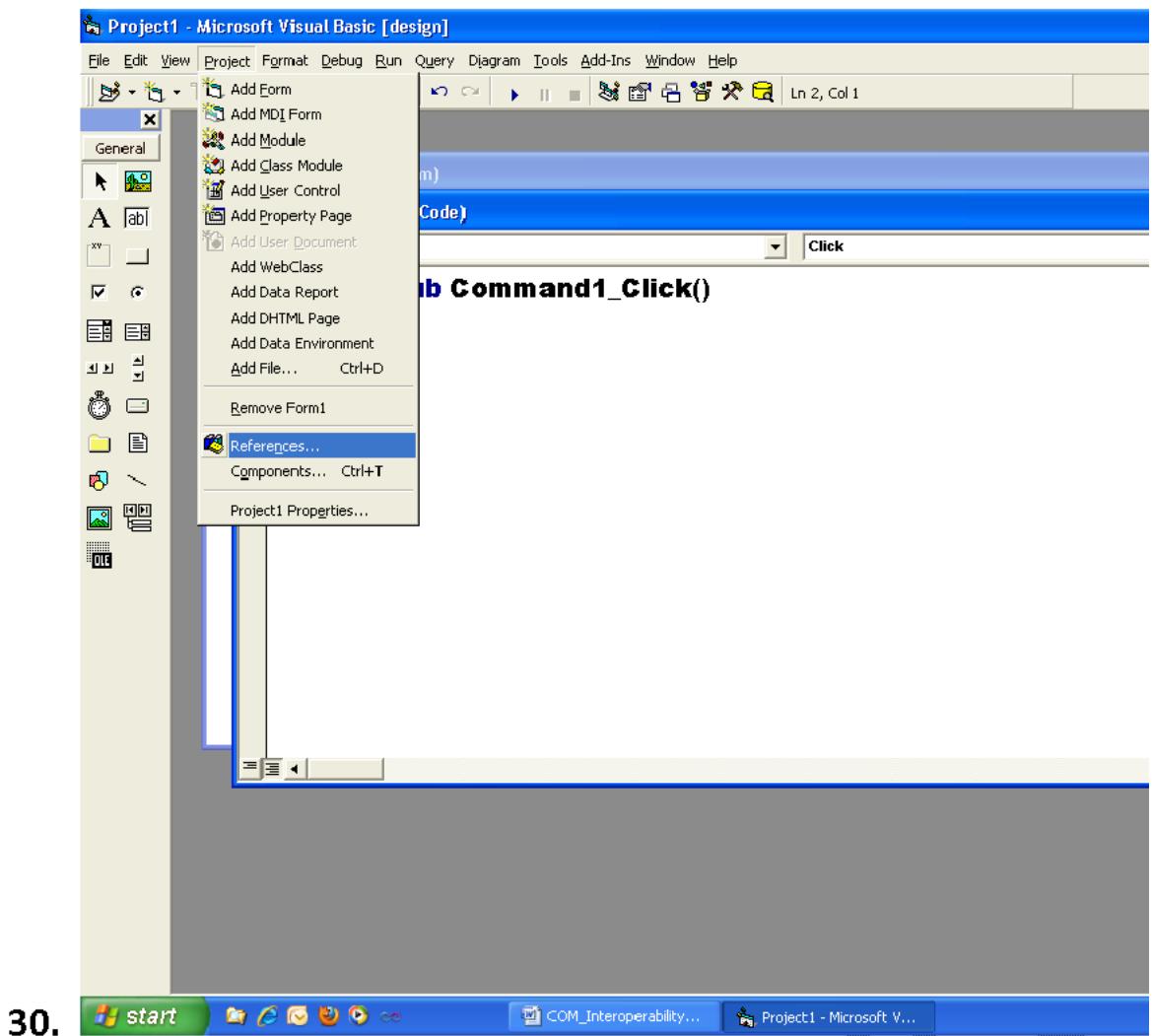




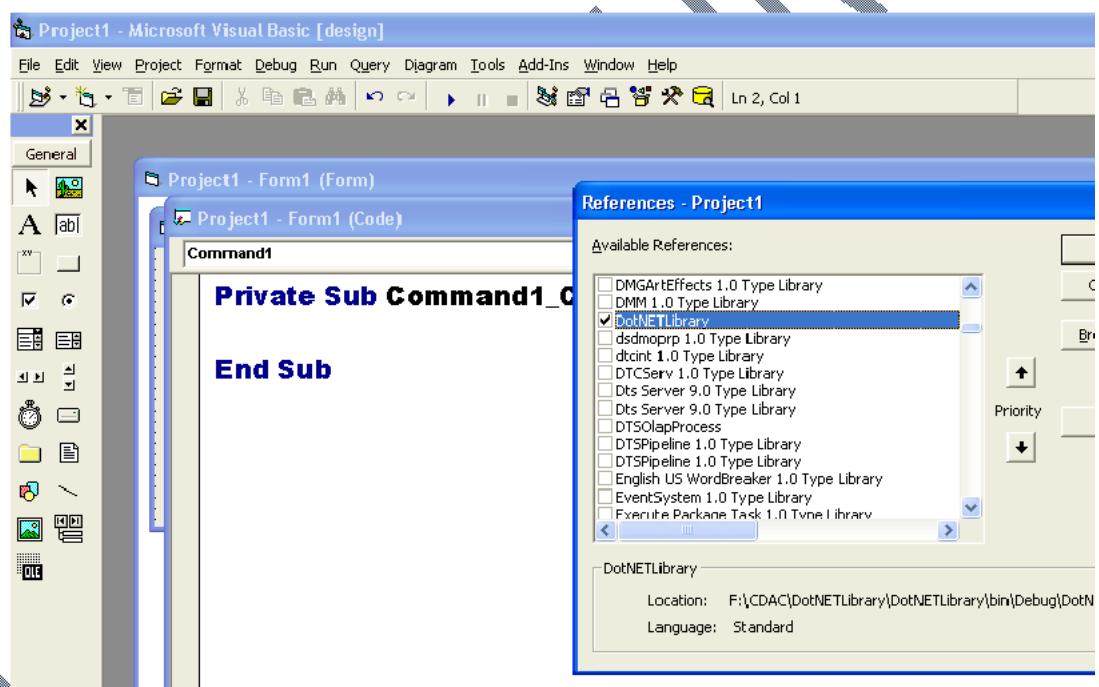


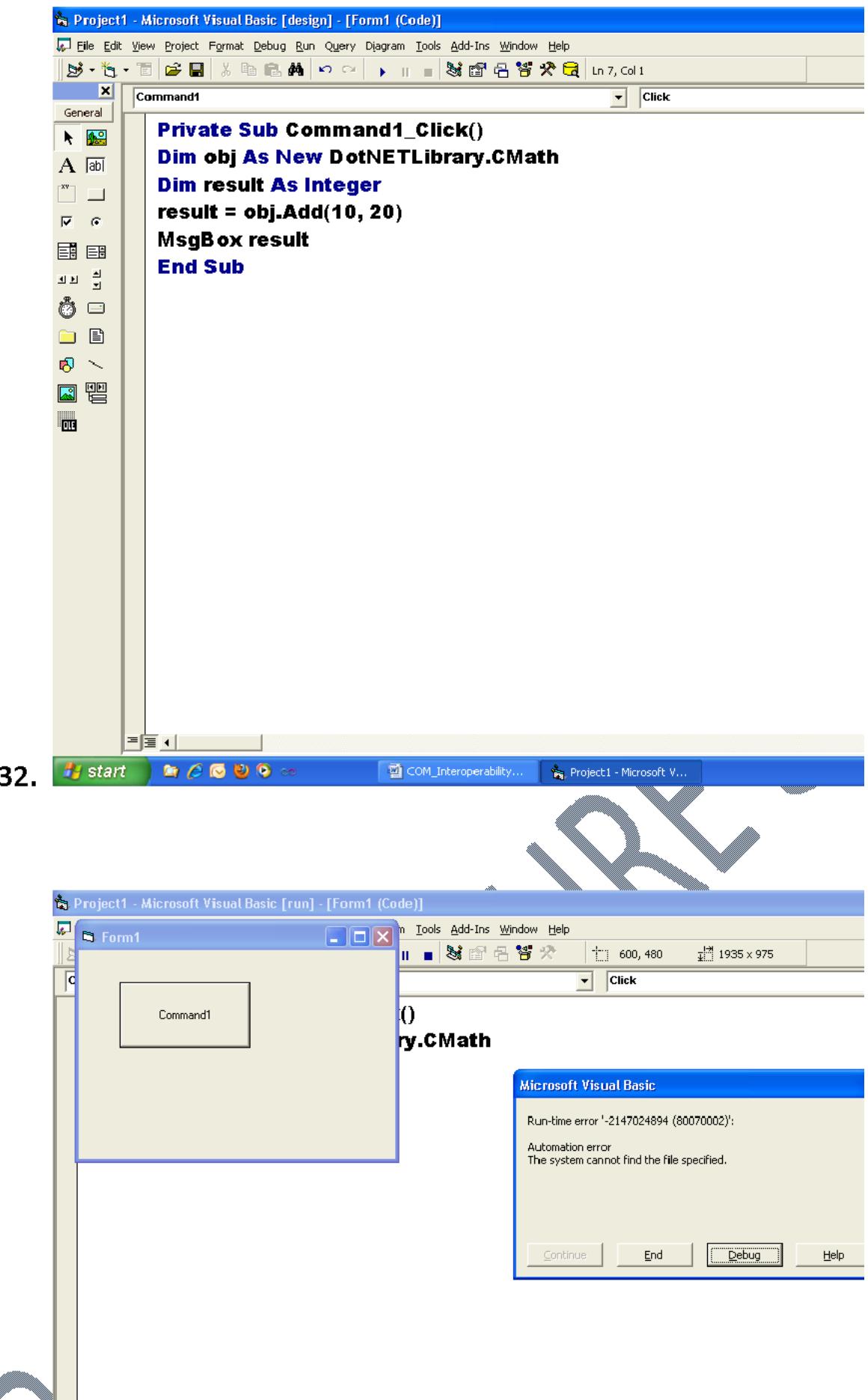
28.

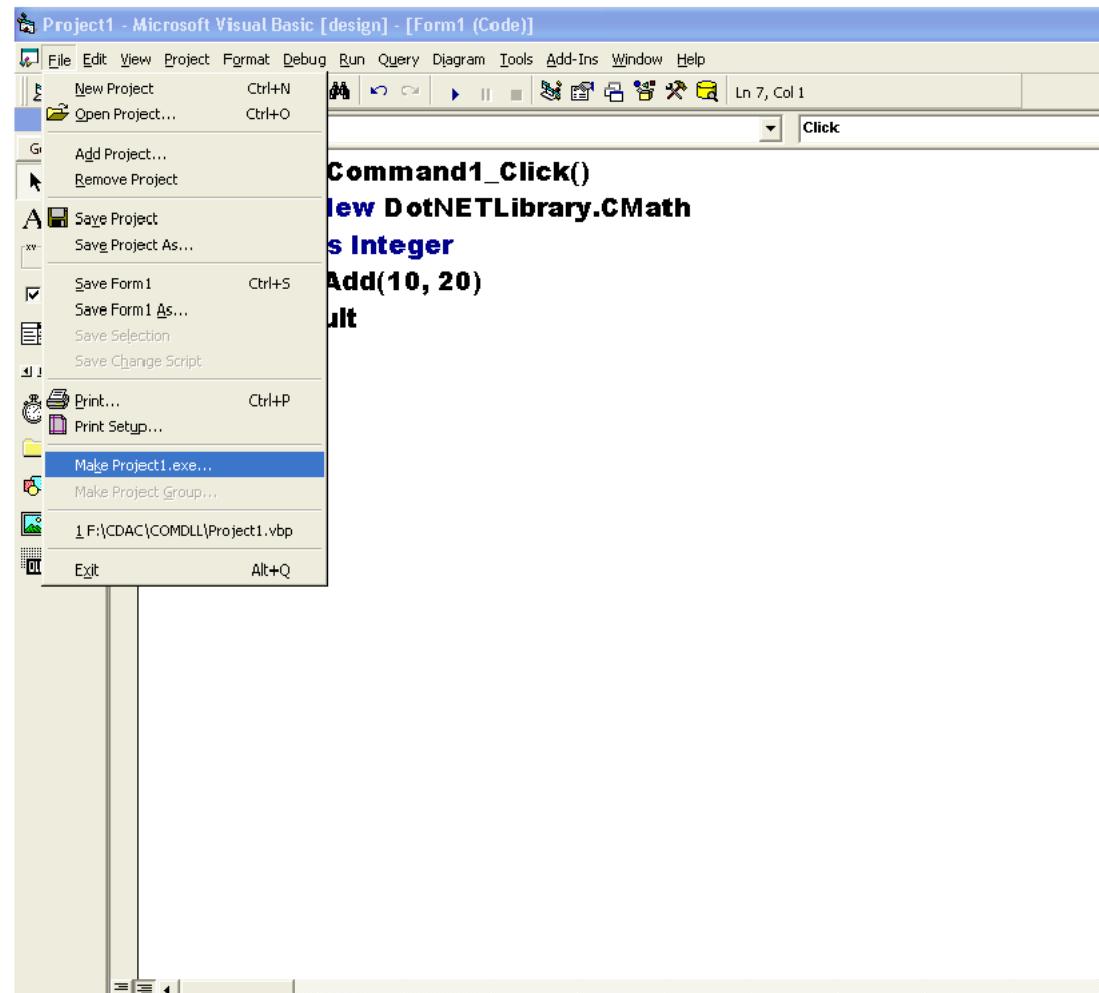




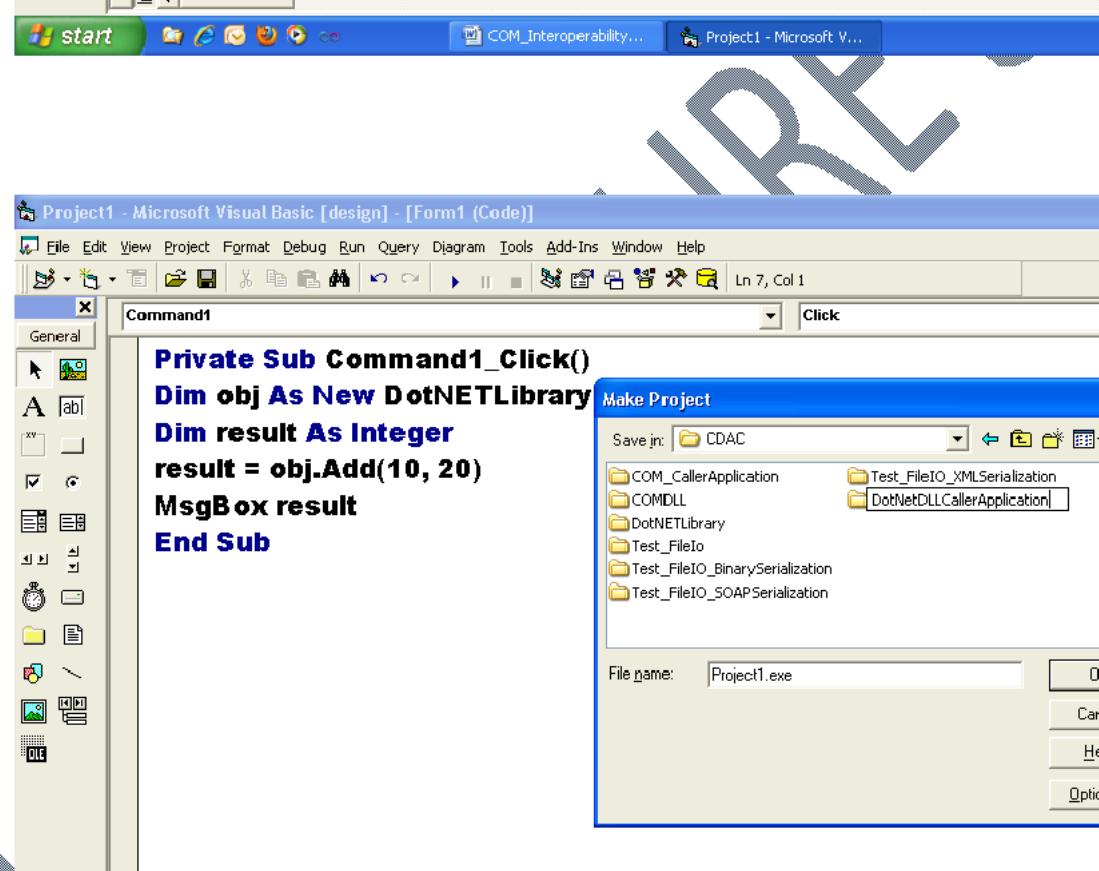
30.

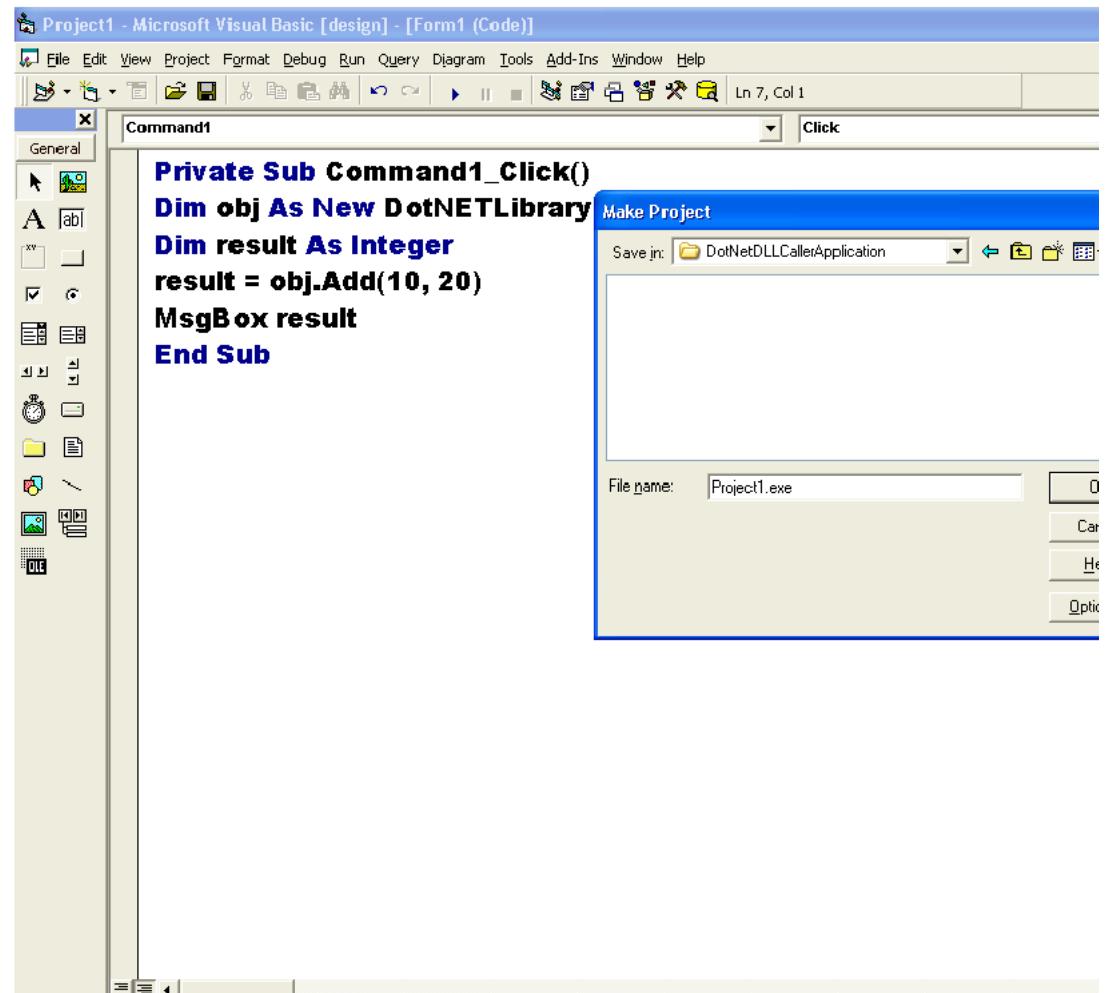




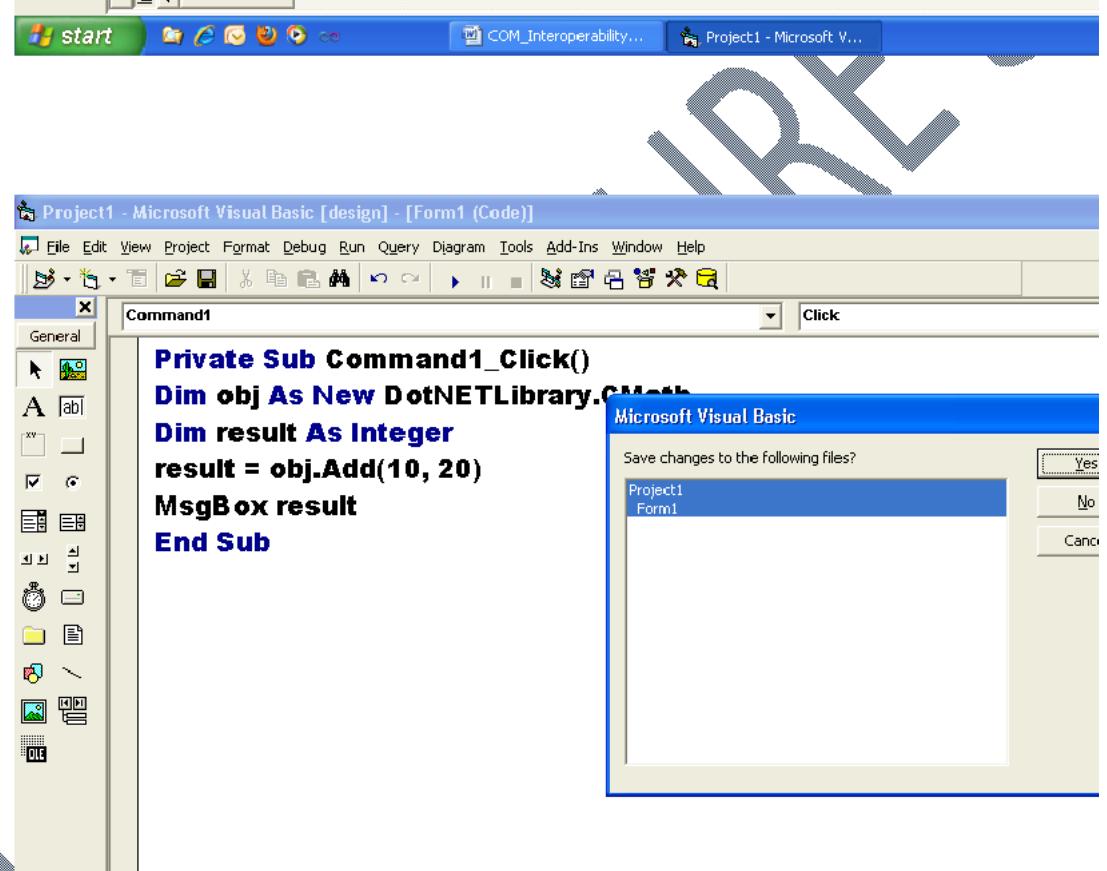


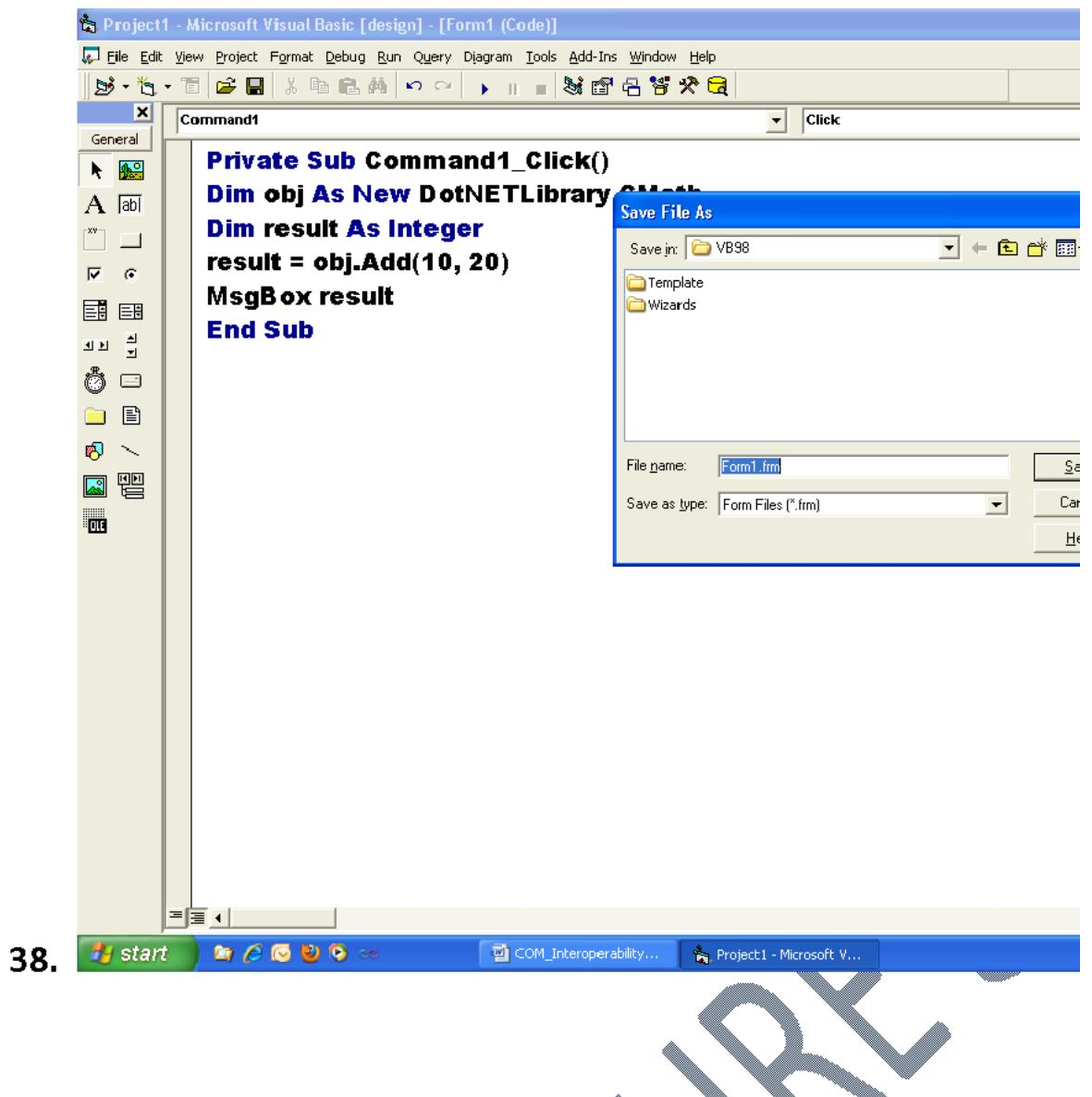
34.



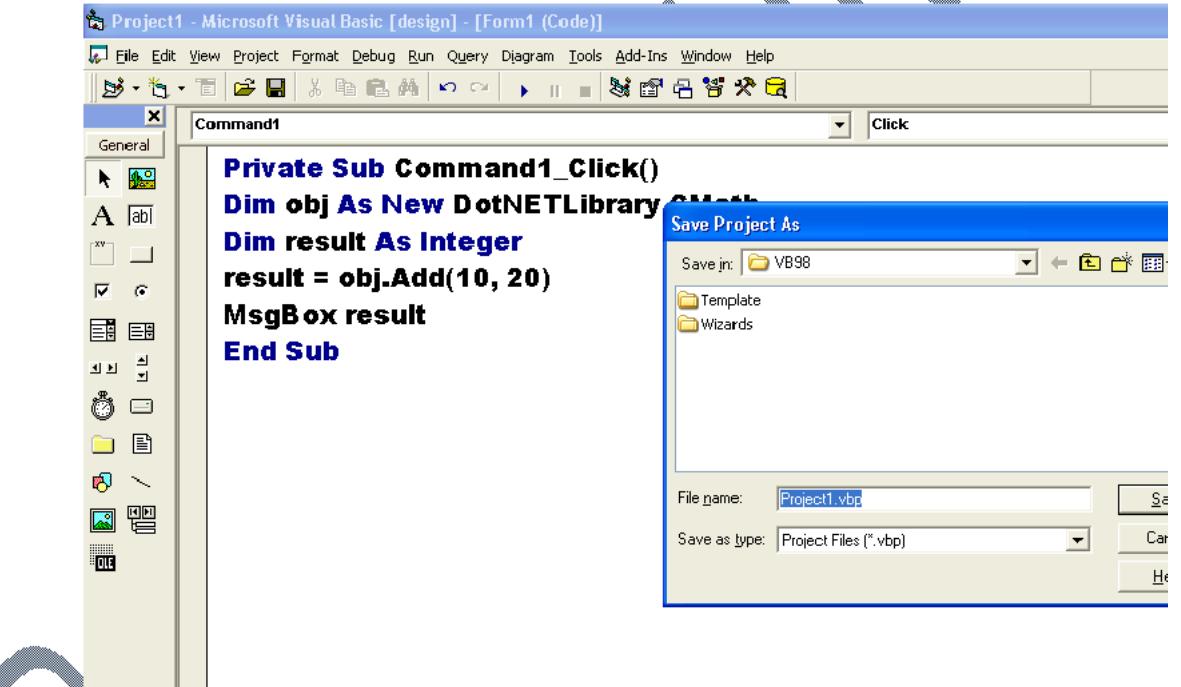


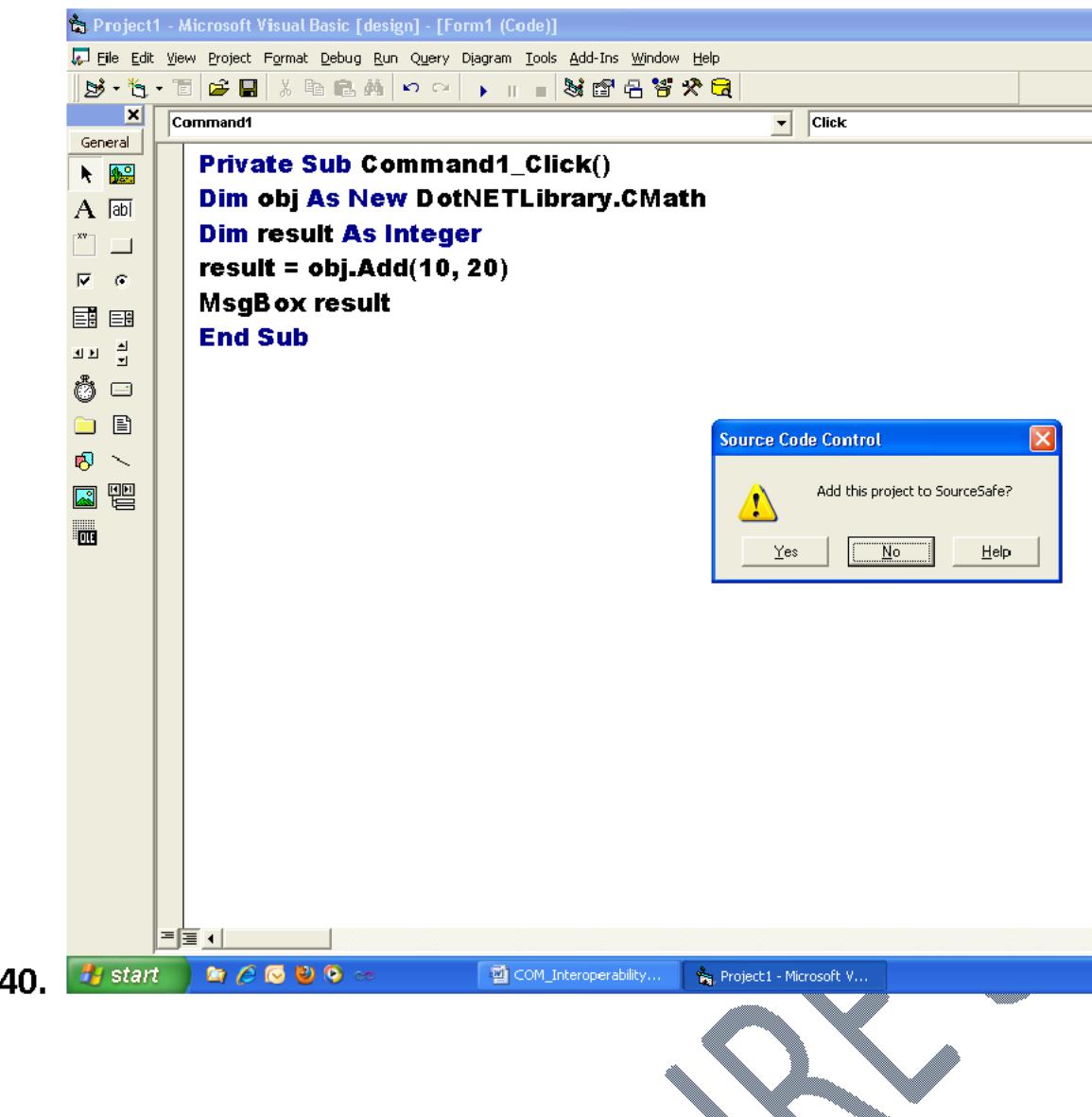
36.



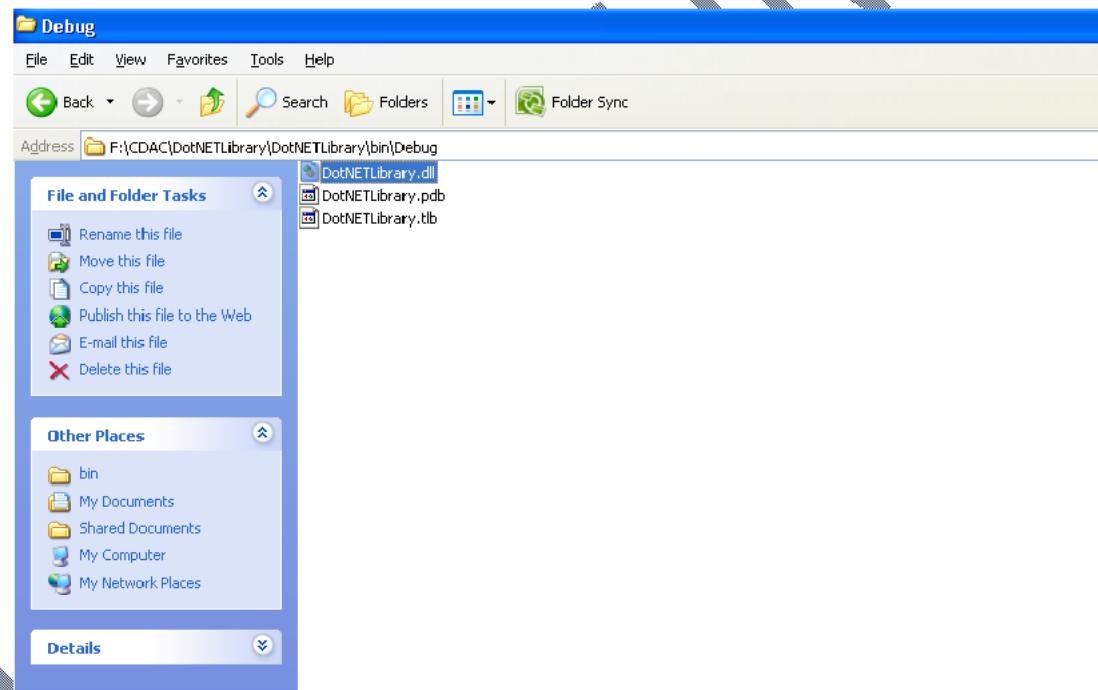


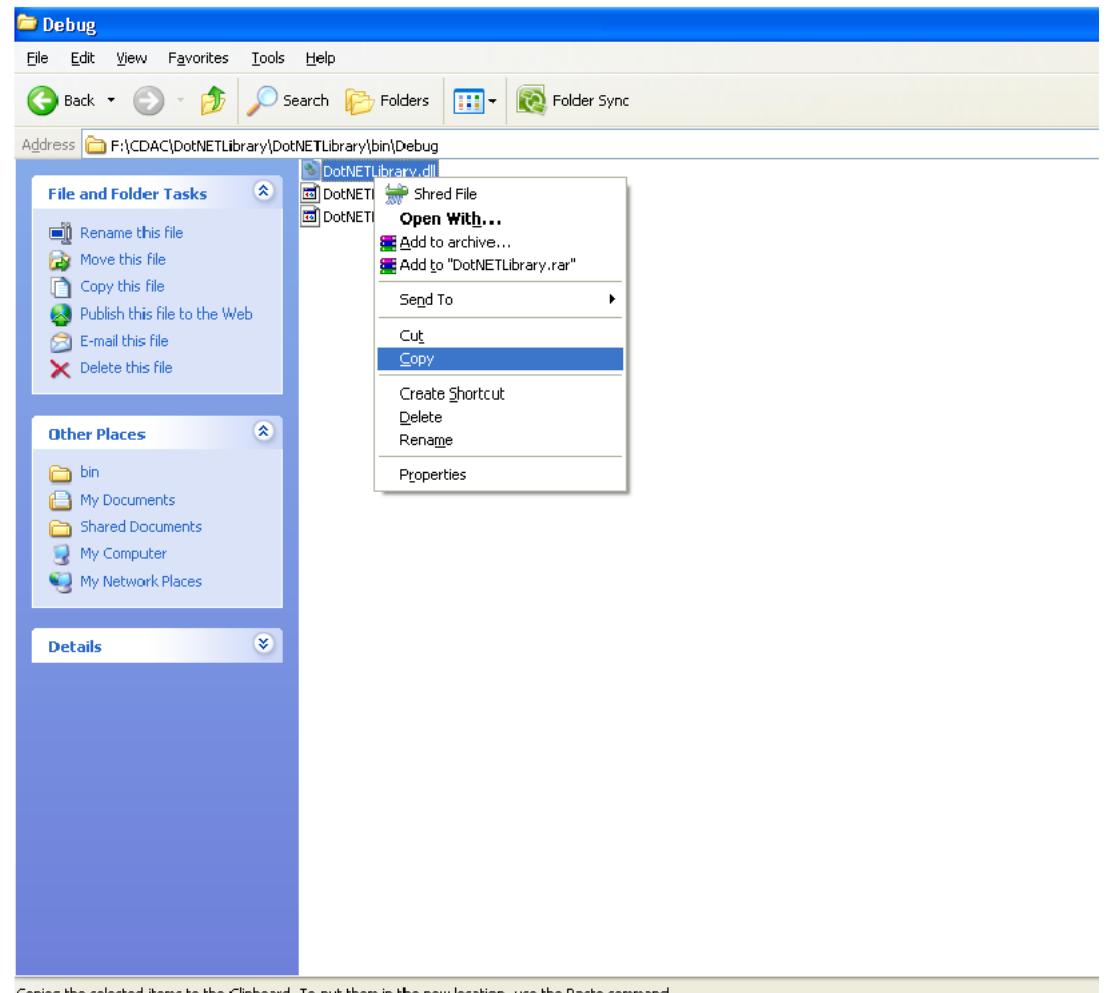
38.





40.





42.

