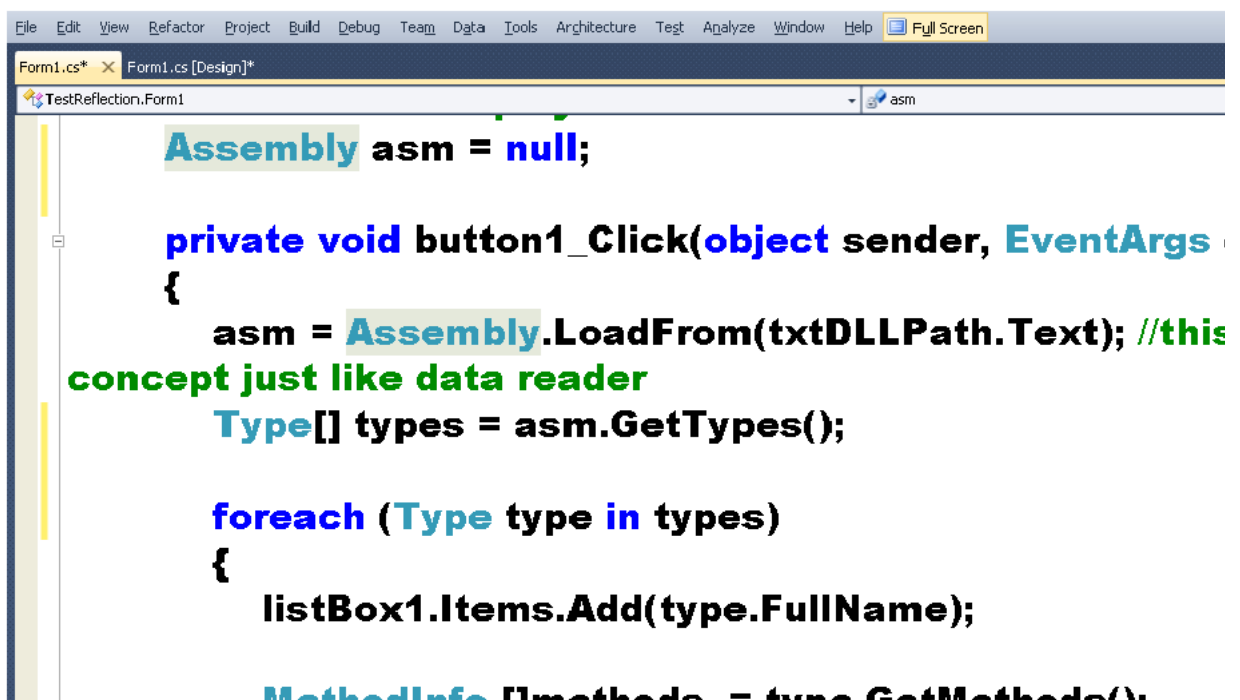


The screenshot shows the Visual Studio IDE with the 'Form1.cs' file open in the 'Design' view. The code editor displays the 'Form1_Load' method, which is a private void method that takes 'object sender' and 'EventArgs e' as parameters. Inside the method, the 'txtDLLPath.Text' property is set to a string representing a file path: '@"G:\CDAC\Reflection\MathDLL\MathDLL.dll"'. The IDE interface includes a menu bar at the top with options like File, Edit, View, Refactor, Project, Build, Debug, Team, Data, Tools, Architecture, Test, Analyze, Window, and Help. A toolbar on the right shows a 'Full Screen' button. The status bar at the bottom indicates 'Ready'.

```
private void Form1_Load(object sender, EventArgs e)
{
    txtDLLPath.Text = @"G:\CDAC\Reflection\MathDLL\MathDLL.dll";
}
```



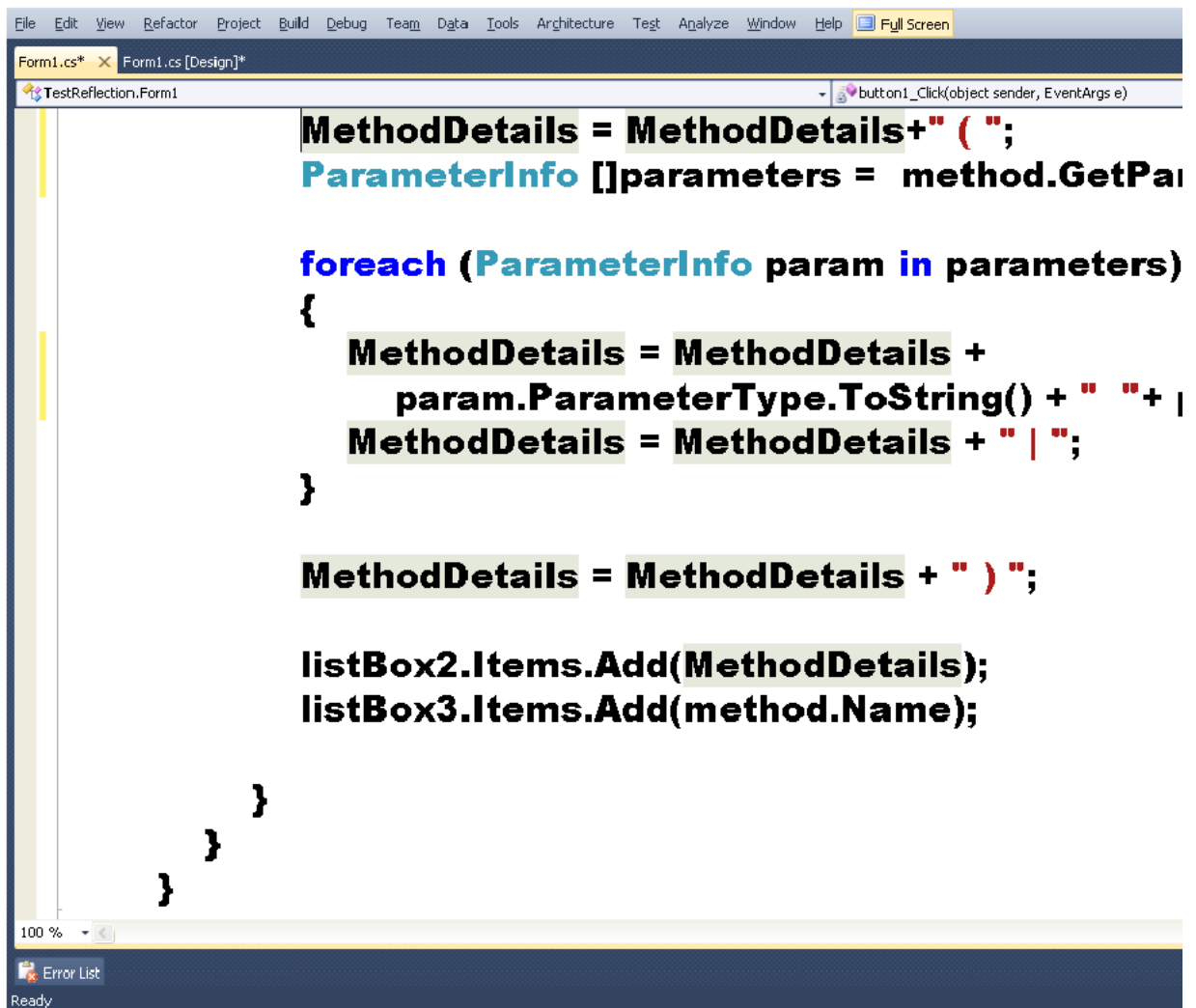
The screenshot shows the Visual Studio IDE with the 'Form1.cs' file open in the 'Design' view. The code editor displays the 'button1_Click' method, which is a private void method that takes 'object sender' and 'EventArgs e' as parameters. Inside the method, an 'Assembly' object named 'asm' is initialized to null. Then, 'asm' is assigned the value of 'Assembly.LoadFrom(txtDLLPath.Text)', with a comment indicating this is a concept just like a data reader. Next, 'types' is assigned the value of 'asm.GetTypes()'. A 'foreach' loop iterates over 'types', and inside the loop, 'listBox1.Items.Add(type.FullName)' is called. Finally, 'MethodInfo[] methods' is assigned the value of 'type.GetMethods()'. The IDE interface is consistent with the previous screenshot, showing the menu bar, toolbar, and status bar.

```
Assembly asm = null;

private void button1_Click(object sender, EventArgs e)
{
    asm = Assembly.LoadFrom(txtDLLPath.Text); //this
    concept just like data reader
    Type[] types = asm.GetTypes();

    foreach (Type type in types)
    {
        listBox1.Items.Add(type.FullName);

        MethodInfo[] methods = type.GetMethods();
    }
}
```



```

File Edit View Refactor Project Build Debug Team Data Tools Architecture Test Analyze Window Help Full Screen
Form1.cs* Form1.cs [Design]*
TestReflection.Form1 button1_Click(object sender, EventArgs e)

MethodDetails = MethodDetails + " ( ";
ParameterInfo []parameters = method.GetParameters();

foreach (ParameterInfo param in parameters)
{
    MethodDetails = MethodDetails +
        param.ParameterType.ToString() + " " + |
    MethodDetails = MethodDetails + " | ";
}

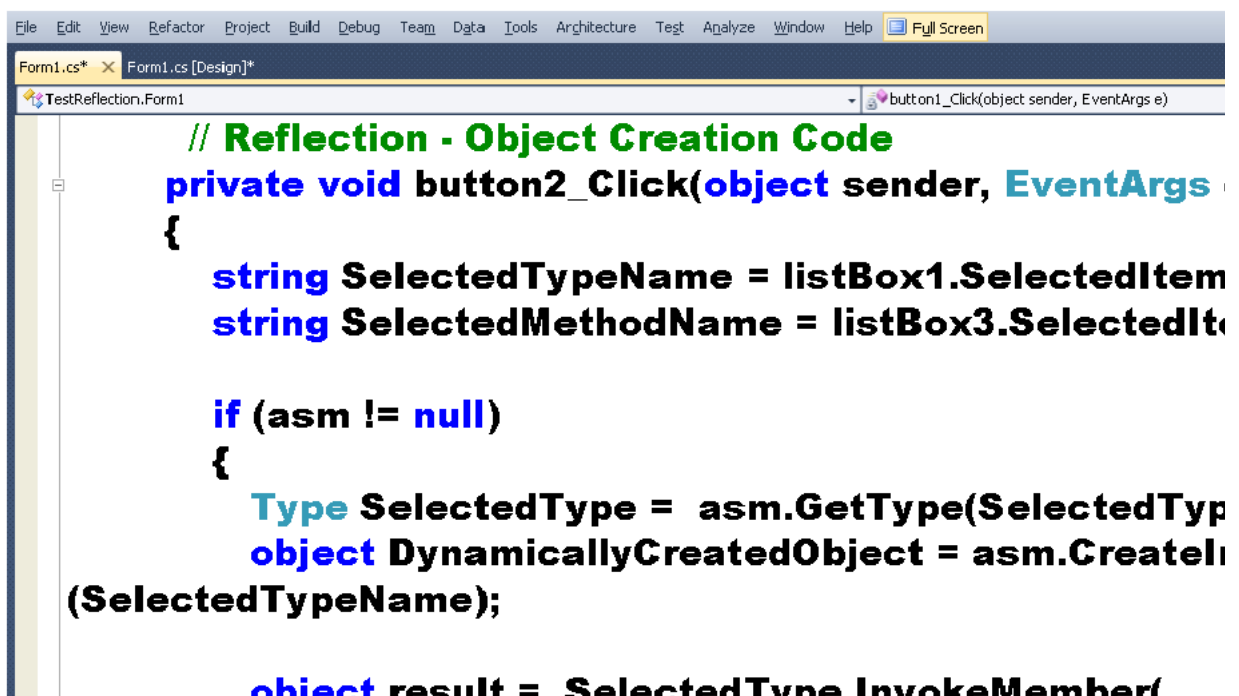
MethodDetails = MethodDetails + " ) ";

listBox2.Items.Add(MethodDetails);
listBox3.Items.Add(method.Name);

}
}
}

100 %
Error List
Ready

```



```

File Edit View Refactor Project Build Debug Team Data Tools Architecture Test Analyze Window Help Full Screen
Form1.cs* Form1.cs [Design]*
TestReflection.Form1 button1_Click(object sender, EventArgs e)

// Reflection - Object Creation Code
private void button2_Click(object sender, EventArgs e)
{
    string SelectedTypeName = listBox1.SelectedItem.ToString();
    string SelectedMethodName = listBox3.SelectedItem.ToString();

    if (asm != null)
    {
        Type SelectedType = asm.GetType(SelectedTypeName);
        object DynamicallyCreatedObject = asm.CreateInstance(
            SelectedTypeName);

        object result = SelectedType.InvokeMember(

```

