International University of Business Agriculture and Technology

Department: Computer Science and Engineering

Semester: Spring 2025

Course Name: Visual Programming

Course Code: CSC 440

Section: A

Lab Report topic: Lab task 02

Submitted To:

Suhala Lamia
Lecturer
Department of Computer Science and Engineering

Submitted By:

Samiul Karim Mazumder
22303308

Date of Submission: 8/03/25

Experiment 01: Demonstrate the use of Public, Private, Protected, Internal, Protected Internal, Private Protected Access Modifiers.

Objective: This lab aims to demonstrate the six different Access Modifiers in C#. These modifiers define the accessibility of variables and methods in a class. The six access modifiers covered are:

- Public
- Private
- Protected
- Internal
- Protected Internal
- Private Protected

Each modifier is implemented in a separate file within the same project.

Algorithm:

1. **Start**
2. Create a **namespace (lab2)** and define six different classes in six separate files, each demonstrating a different access modifier.
3. Implement the following access modifiers:
    - **Public:** Accessible everywhere.
    - **Private:** Accessible only inside the same class.
    - **Protected:** Accessible inside the same class and derived classes.
    - **Internal:** Accessible within the same assembly.
    - **Protected Internal:** Accessible within the same assembly and derived classes in other assemblies.
    - **Private Protected:** Accessible only in the same class and derived classes within the same assembly.
4. Implement a **Main program (Program.cs)** that creates instances of these classes and calls their methods where allowed.
5. **Run the program** to observe the access restrictions.
6. **End**

Program:

1. Public Access Modifier

```
namespace lab2
{
   public class PublicEx
   {
      public string message = "This is a public variable";

      public void ShowMessage()
      {
         Console.WriteLine(message);
      }
   }
}
```

2. Private Access Modifier

```
namespace lab2
{
   class PrivateEx
   {
      private string message = "This is a private variable";

      private void ShowMessage()
      {
         Console.WriteLine(message);
      }

      public void AccessPrivateMethod()
      {
         ShowMessage();
      }
```

```
    }
}
```

3. Protected Access Modifier

```
namespace lab2
{
    class BaseProtected
    {
        protected string message = "This is a protected variable";
    }

    class ProtectedEx : BaseProtected
    {
        public void ShowMessage()
        {
            Console.WriteLine(message);
        }
    }
}
```

4. Internal Access Modifier

```
namespace lab2
{
    class InternalEx
    {
        internal string message = "This is an internal variable";

        public void ShowMessage()
        {
            Console.WriteLine(message);
        }
    }
}
```

5. Protected Internal Access Modifier

```
namespace lab2
{
   class BaseProtectedInternal
   {
      protected internal string message = "This is a protected internal variable";
   }

   class ProtectedInternalEx : BaseProtectedInternal
   {
      public void ShowMessage()
      {
         Console.WriteLine(message);
      }
   }
}
```

6. Private Protected Access Modifier

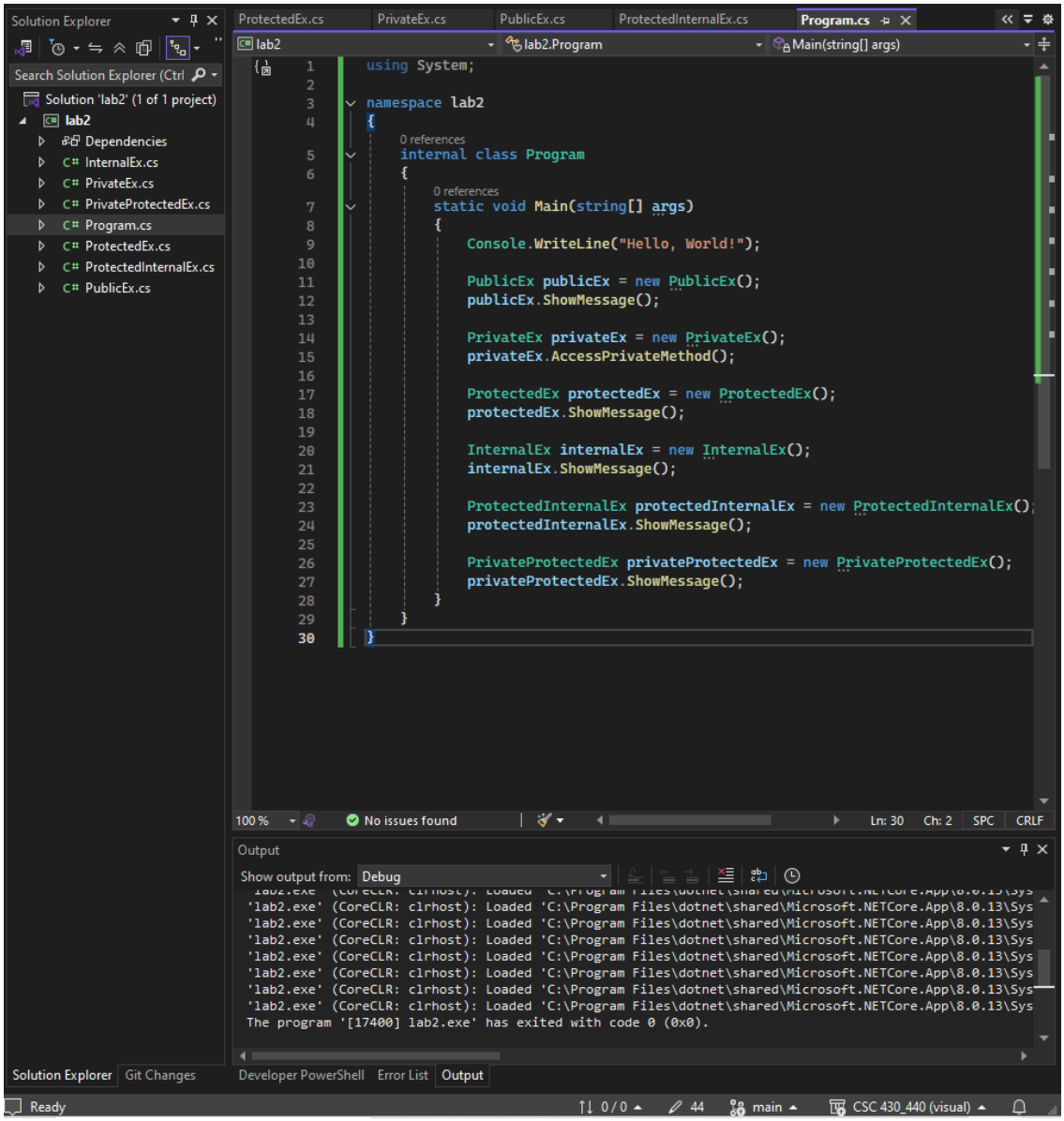
```
namespace lab2
{
   class BasePrivateProtected
   {
      private protected string message = "This is a private protected variable";
   }

   class PrivateProtectedEx : BasePrivateProtected
   {
      public void ShowMessage()
      {
         Console.WriteLine(message);
      }
   }
}
```

 # Main Program

```
using System;

namespace lab2
{
   internal class Program
   {
      static void Main(string[] args)
      {
         Console.WriteLine("Hello, World!");

         PublicEx publicEx = new PublicEx();
         publicEx.ShowMessage();

         PrivateEx privateEx = new PrivateEx();
         privateEx.AccessPrivateMethod();

         ProtectedEx protectedEx = new ProtectedEx();
         protectedEx.ShowMessage();

         InternalEx internalEx = new InternalEx();
         internalEx.ShowMessage();

         ProtectedInternalEx protectedInternalEx = new ProtectedInternalEx();
         protectedInternalEx.ShowMessage();

         PrivateProtectedEx privateProtectedEx = new PrivateProtectedEx();
         privateProtectedEx.ShowMessage();
      }
   }
}
```

Output:


Microsoft Visual Studio Debug Console
Hello, World!
This is a public variable
This is a private variable
This is a protected variable
This is an internal variable
This is a protected internal variable
This is a private protected variable
C:\Users\Puri_Sama\Documents\Uni works\7th semester\CSC 430_440 (visual)\lab2\bin\Debug\net8.0\lab2.exe (process 17400) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .


Solution Explorer
Search Solution Explorer (Ctrl
Solution 'lab2' (1 of 1 project)
lab2
Dependencies
InternalEx.cs
PrivateEx.cs
PrivateProtectedEx.cs
Program.cs
ProtectedEx.cs
ProtectedInternalEx.cs
PublicEx.cs
ProtectedEx.cs
PrivateEx.cs
PublicEx.cs
ProtectedInternalEx.cs
Program.cs
lab2
lab2.Program
Main(string[] args)
using System;
namespace lab2
{
0 references
internal class Program
{
0 references
static void Main(string[] args)
{
Console.WriteLine("Hello, World!");
PublicEx publicEx = new PublicEx();
publicEx.ShowMessage();
PrivateEx privateEx = new PrivateEx();
privateEx.AccessPrivateMethod();
ProtectedEx protectedEx = new ProtectedEx();
protectedEx.ShowMessage();
InternalEx internalEx = new InternalEx();
internalEx.ShowMessage();
ProtectedInternalEx protectedInternalEx = new ProtectedInternalEx();
protectedInternalEx.ShowMessage();
PrivateProtectedEx privateProtectedEx = new PrivateProtectedEx();
privateProtectedEx.ShowMessage();
}
}
}
100 %
No issues found
Ln: 30
Ch: 2
SPC
CRLF
Output
Show output from: Debug
'lab2.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.13\Sys
The program '[17400] lab2.exe' has exited with code 0 (0x0).
Solution Explorer
Git Changes
Developer PowerShell
Error List
Output
Ready
0 / 0
44
main
CSC 430_440 (visual)