01_DATA_TYPES\08_STATIC_DEMO\Program.cs

```csharp
using System;

namespace _08_STATIC_DEMO
{
    public class Counter
    {
        // Static field: shared across all instances
        public static int StaticCount = 0;

        // Non-static (instance) field: unique to each instance
        public int InstanceCount = 0;

        // Static method: can only access static members
        public static void IncrementStaticCount()
        {

            StaticCount++;
            //InstanceCount++; // An object reference is required for the non-static field,
method, or property 'Counter.InstanceCount'

            Console.WriteLine($"Static Count: {StaticCount}");
            // Cannot access InstanceCount here because it's non-static
        }

        // Non-static (instance) method: can access both static and non-static members
        public void IncrementInstanceCount()
        {
            InstanceCount++;
            StaticCount++; // Can access static members
            Console.WriteLine($"Instance Count: {InstanceCount}, Static Count:
{StaticCount}");
        }
    }

    class Program
    {

        static void Main(string[] args)
        {
            // Accessing static member without creating an instance
            Console.WriteLine("Initial Static Count: " + Counter.StaticCount); // Output: 0
            Counter.IncrementStaticCount(); // Output: Static Count: 1

            // Creating instances of Counter
            Counter counter1 = new Counter();
            Counter counter2 = new Counter();

            // Calling non-static method on instances
            counter1.IncrementInstanceCount(); // Output: Instance Count: 1, Static Count:
2
            counter2.IncrementInstanceCount(); // Output: Instance Count: 1, Static Count:
3
```

```
49
50            // Static count is shared, instance count is unique
51            Console.WriteLine($"counter1 Instance Count: {counter1.InstanceCount}"); //
   Output: 1
52            Console.WriteLine($"counter2 Instance Count: {counter2.InstanceCount}"); //
   Output: 1
53            Console.WriteLine($"Final Static Count: {Counter.StaticCount}"); // Output: 3
54        }
55    }
56 }
57
```