Session: 10

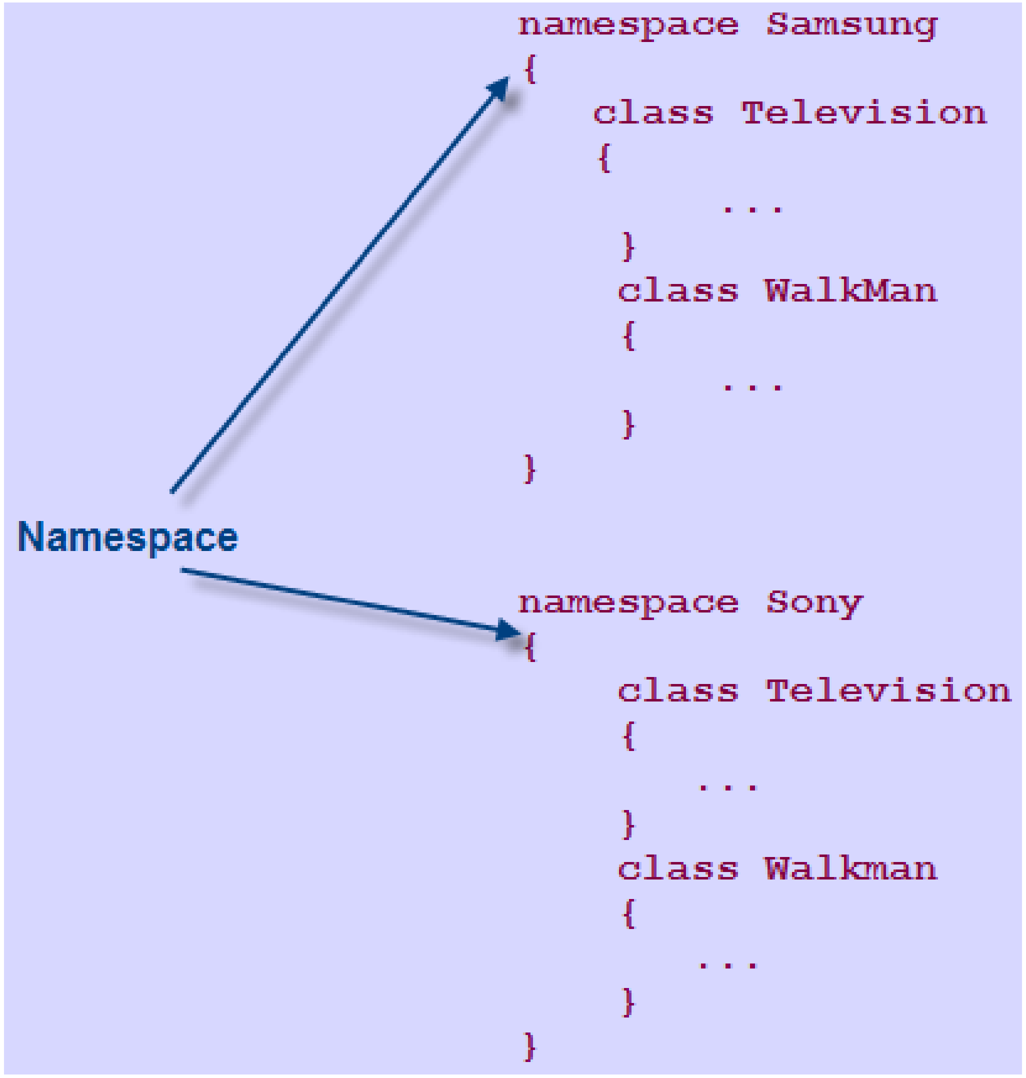
Namespaces

Objectives

- Define and describe namespaces
- Explain nested namespaces

Namespaces

- A namespace:
  - Is used to group classes logically and prevent name clashes between classes with identical names.
  - Reduces any complexities when the same program is required in another application.



```
namespace Samsung
{
   class Television
   {
        . . .
    }
    class WalkMan
    {
        . . .
    }
}

namespace Sony
{
    class Television
    {
       . . .
    }
    class Walkman
    {
       . . .
    }
}
```

Using Namespaces

- allows to specify a unique identifier for each namespace.
- helps you to access the classes within the namespace.
- apart from classes, the following structures can be declared in a namespace:

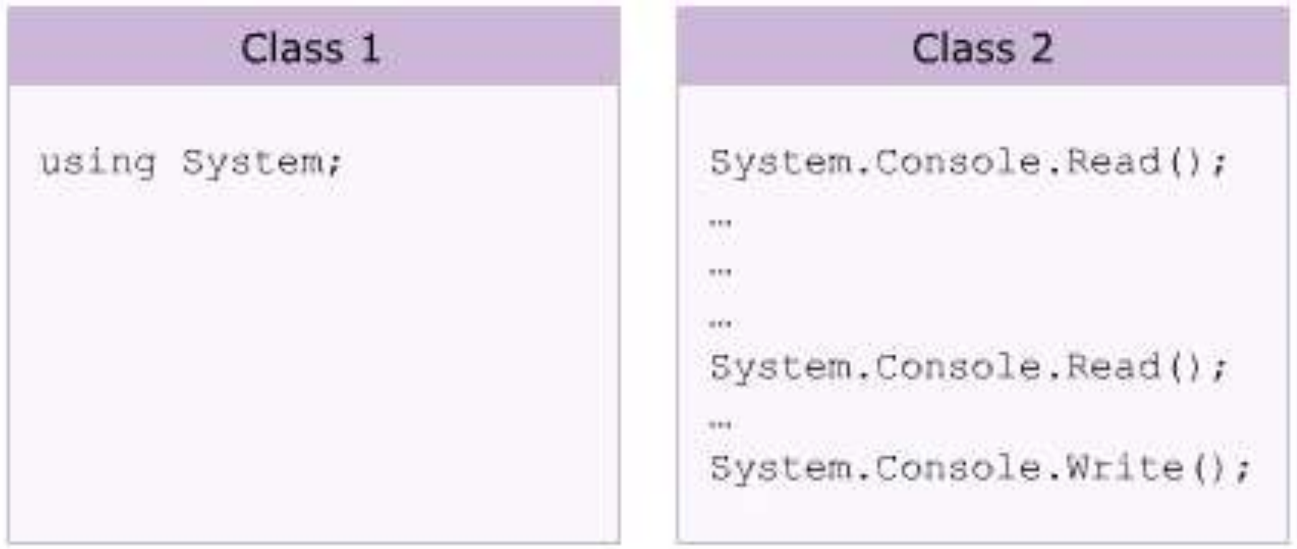
Interface	• is a reference type that contains declarations of the events, indexers, methods, and properties.
Structure	• is a value type • can include fields, methods, constants, constructors, properties, indexers, operators, and other structures.
Enumeration	• is a value type that consists of a list of named constants.
Delegate	• is a reference type that refers to one or more methods. • can be used to pass data as parameters to methods.

Built-in Namespaces

`System.Collections`	• contains classes and interfaces that define complex data structures such as lists, queues, bit arrays, hash tables, and dictionaries.
`System.Data`	• contains classes that make up the ADO.NET architecture. • The ADO.NET architecture allows to build components that can be used to insert, modify, and delete data from multiple data sources.
`System.Diagnostics`	• contains classes that are used to interact with the system processes. • also provides classes that are used to debug applications and trace the execution of the code.
`System.IO`	• contains classes that enable you to read from and write to data streams and files.
`System.Web`	• contains classes that allow you to create Web-based applications.
`System.Net`	• provides classes and interfaces that allow communication between the browser and the server.

Using the System Namespace

- The two approaches of referencing the `System` namespace are:



```
using System;
```

```
System.Console.Read();
...
...
...
System.Console.Read();
...
System.Console.Write();
```

Efficient Programming **Inefficient Programming**

- Though both are technically valid, the first approach is more recommended.

Using the system-defined Namespace

- The following syntax is used to access a method in a system-defined namespace:

```
<NamespaceName>.<ClassName>.<MethodName>;
```

- In the syntax:
    - `NamespaceName`: Is the name of the namespace.
    - `ClassName`: Is the name of the class that you want to access.
    - `MethodName`: Is the name of the method within the class that is to be invoked.

Custom Namespaces

Snippet

```
using System;
namespace Automotive
{
   public class SpareParts {
      string _spareName;
      public SpareParts() {
        _spareName = "Gear Box";
      }
      public void Display() {
        Console.WriteLine("Spare Part name: _spareName);
      }
   }
}
```

Unqualified Naming

```
using System;
using Students;
namespace Students {
   class StudentDetails {
      string _studName = “Alexander”;
      int _studID = 30;
      public StudentDetails() {
         Console.WriteLine(“Student Name: “ + _studName);
         Console.WriteLine(“Student ID: “ + _studID);
      }
   }
}
namespace Examination {
   class ScoreReport {
      public string Subject = “Science”;
      public int Marks = 60;
      static void Main(string[] args) {
         StudentDetails objStudents = new StudentDetails();
         ScoreReport objReport = new ScoreReport();
         Console.WriteLine(“Subject: “ + objReport.Subject);
         Console.WriteLine(“Marks: “ + objReport.Marks);
      }
   }
}
```

```
using System;
namespace Students {
   class StudentDetails {
      string _studName = “Alexander”;
      int _studId = 30;
      public StudentDetails() {
         Console.WriteLine(“Student Name: “ + _studName);
         Console.WriteLine(“Student ID: “ + _studId);
      }
   }
}
namespace Examination {
   class ScoreReport {
      public string Subject = “Science”;
      public int Marks = 60;
      static void Main(string[] args) {
         Students.StudentDetails os = new Students.StudentDetails();
         ScoreReport or = new ScoreReport();
         Console.WriteLine(“Subject: “ + or.Subject);
         Console.WriteLine(“Marks: “ + or.Marks);
      }
   }
}
```

Implementing Nested Namespaces

- C# allows to create a hierarchy of namespaces by creating namespaces within namespaces.
- Such nesting of namespaces is done by enclosing one namespace declaration inside the declaration of the other namespace.

Syntax

```
namespace Contact
{
    public class Employees
    {
      public int EmpID;
    }
    namespace Salary
    {
      public class SalaryDetails
      {
         public double EmpSalary;
      }
    }
}
```

Namespace Aliases

```
namespace Bank.Accounts.EmployeeDetails
{
    public class Employees {
        public string EmpName;
     }
}
```

```
using IO = System.Console;
using Emp = Bank.Accounts.EmployeeDetails;
class AliasExample
{
   static void Main (string[] args)
   {
      Emp.Employees objEmp = new Emp.Employees();
      objEmp.EmpName = "Peter";
      IO.WriteLine("Employee Name: " + objEmp.EmpName);
   }
}
```

Summary

- A namespace in C# is used to group classes logically and prevent name clashes between classes with identical names.
- The System namespace is imported by default in the .NET Framework.
- Custom namespaces enable you to control the scope of a class by deciding the appropriate namespace for the class.
- You cannot apply access modifiers such as public, protected, private, or internal to namespaces.
- A class outside its namespace can be accessed by specifying its namespace followed by the dot operator and the class name.
- C# supports nested namespaces that allows you to define namespaces within a namespace.
- External aliasing in C# allows the users to define assembly qualified namespace aliases.