

.NET TECHNOLOGIES



Discover apps before they launch.
A curated list of apps coming soon.
Find apps that are trending.
"Spring new" from apps.





.NET Technologies - Lecture 4

Presented by: Drashti Shrimal

UNIT 1: Topics in this presentation

.NET concepts:

- Methods**
- Classes**
- Namespaces and Assemblies**
- Static Members**
- Casting Objects**
- Partial Classes**



Methods

- A method is a code block that contains a series of statements.
- A program causes the statements to be executed by calling the method and specifying any required method arguments.
- In C#, every executed instruction is performed in the context of a method.
- The Main method is the entry point for every C# application and it's called by the common language runtime (CLR) when the program is started.

Methods

```
class NumberManipulator {  
  
    public int FindMax(int num1, int  
num2) {  
        /* local variable declaration */  
        int result;  
  
        if (num1 > num2)  
            result = num1;  
  
        else  
            result = num2;  
  
        return result;  
    }  
    ...  
}
```

Classes

- A class is a reference type.
- When an object of the class is created, the variable to which the object is assigned holds only a reference to that memory.
- When the object reference is assigned to a new variable, the new variable refers to the original object.
- Changes made through one variable are reflected in the other variable because they both refer to the same data.

```
class Program
```

```
{
```

```
    static void Main(string[] args)
```

```
{
```

```
        System.Console.WriteLine("Hello  
World!");
```

```
    }
```

```
}
```

Namespaces

- A namespace simply provides a named group of classes, structures, enumerations, delegates, interfaces and other namespaces.
- Within the namespace, all declared items must be uniquely named.
- However, the same name may be duplicated in different namespaces.

Defining Namespace

```
namespace namespace_name  
{  
    // code declarations  
}
```

Syntax to use a name space:

```
namespace_name.item_name;
```

Example:

```
using System;
```

```
using System;
```

```
namespace first_space {  
    class namespace_cl {  
        public void func() {  
            Console.WriteLine("Inside  
first_space");  
        }  
    }  
}
```


Assemblies

- The .NET assembly is the standard for components developed with the Microsoft.NET.
- Dot NET assemblies may or may not be executable, i.e., they might exist as the executable (.exe) file or dynamic link library (DLL) file.
- All the .NET assemblies contain the definition of types, versioning information for the type, meta-data, and manifest. The designers of .NET have worked a lot on the component (assembly) resolution.

Static Class

- A static class is basically the same as a non-static class, but there is one difference: a static class cannot be instantiated. In other words, you cannot use the new operator to create a variable of the class type.
- Because there is no instance variable, you access the members of a static class by using the class name itself.

Casting Objects

- Casting is taking an object and attempting to "force" it to change types.
- When a cast is attempted, if the value of the object is allowable in the new type, the object will be casted into an object of the specified type.
- We cast a value by placing the targeted type in parentheses () next to the value we want to cast.

Partial Classes

- A partial class is a special feature of C#. It provides a special ability to implement the functionality of a single class into multiple files and all these files are combined into a single class file when the application is compiled.
- A partial class is created by using a partial keyword. This keyword is also useful to split the functionality of methods, interfaces, or structure into multiple files.

Partial Classes

```
public partial Clas_name  
{  
    // code  
}
```

Partial Classes

- When you want to chop the functionality of the class, method, interface, or structure into multiple files, then you should use partial keyword and all the files are mandatory to be available at compile time for creating the final file.
- The partial modifier can only present instantly before the keywords like struct, class, and interface.
- Every part of the partial class definition should be in the same assembly and namespace, but you can use a different source file name.

Partial Classes

- Every part of the partial class definition should have the same accessibility as private, protected, etc.
- If any part of the partial class is declared as an abstract, sealed, or base, then the whole class is declared of the same type.
- The user is also allowed to use nested partial types.
- Dissimilar parts may have dissimilar base types, but the final type must inherit all the base types.

Partial Classes

