Introduction To C#

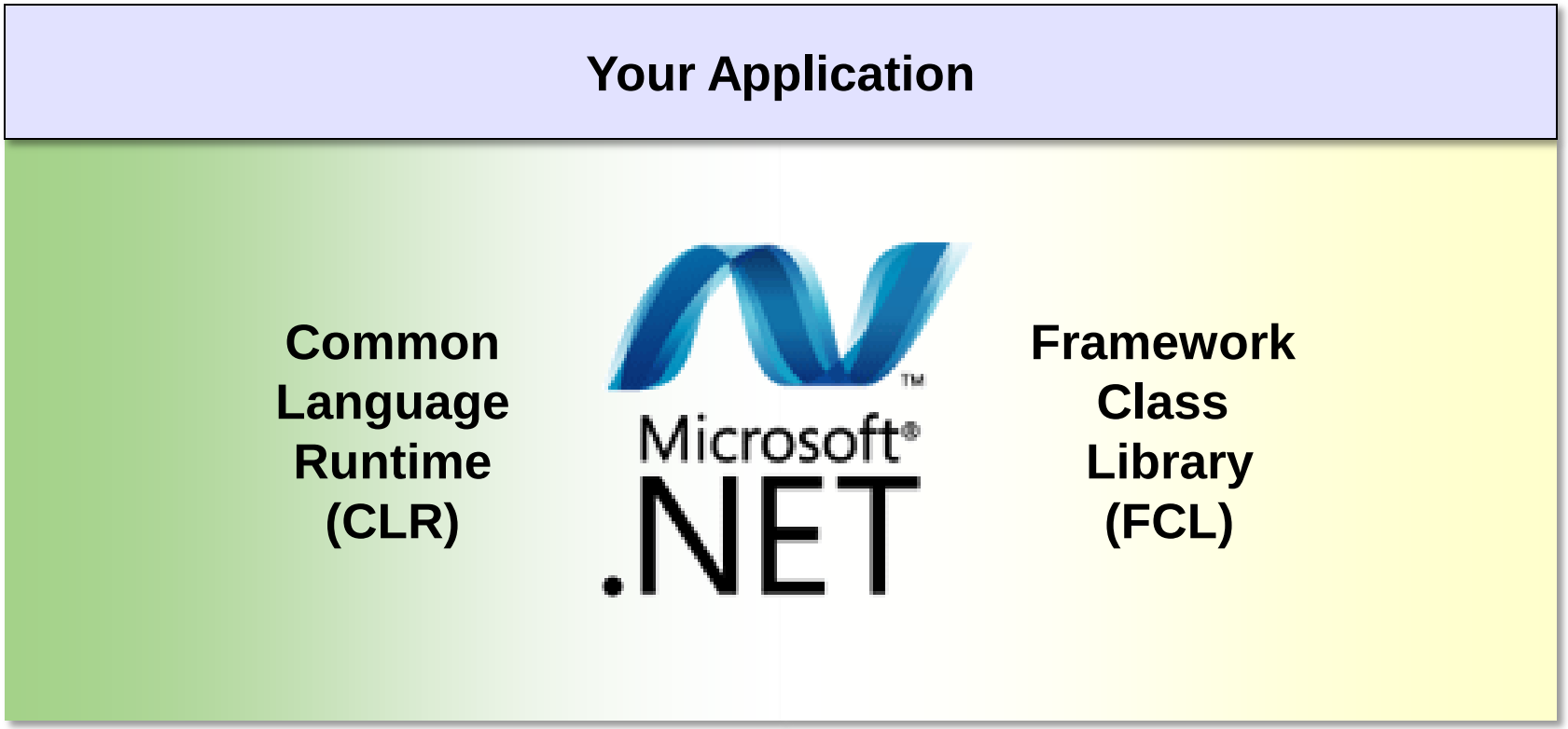
Hello World!

Overview

- What is .NET?
- What is the FCL?
- What is the CLR?
- Building Hello, World!
- Basic expressions and operators
- Compilers and command line tools
- Creating projects with Visual Studio

.NET

- **A software framework**
  - http://msdn.microsoft.com/en-us/netframework/default.aspx



CLR

- **The CLR manages your application when it runs**
  - Memory management
  - Security
  - Operating system and hardware independence
  - Language independence

Common Language Runtime (CLR)

FCL

- **Framework class library**
  - A library of functionality to build applications

Microsoft®
.NET

Framework Class Library (FCL)

What is C#?

- **A standardized language to create .NET components**
  - Standardized by ECMA
  - Create applications, services, and reusable libraries
  - Syntax is similar to Java and C++

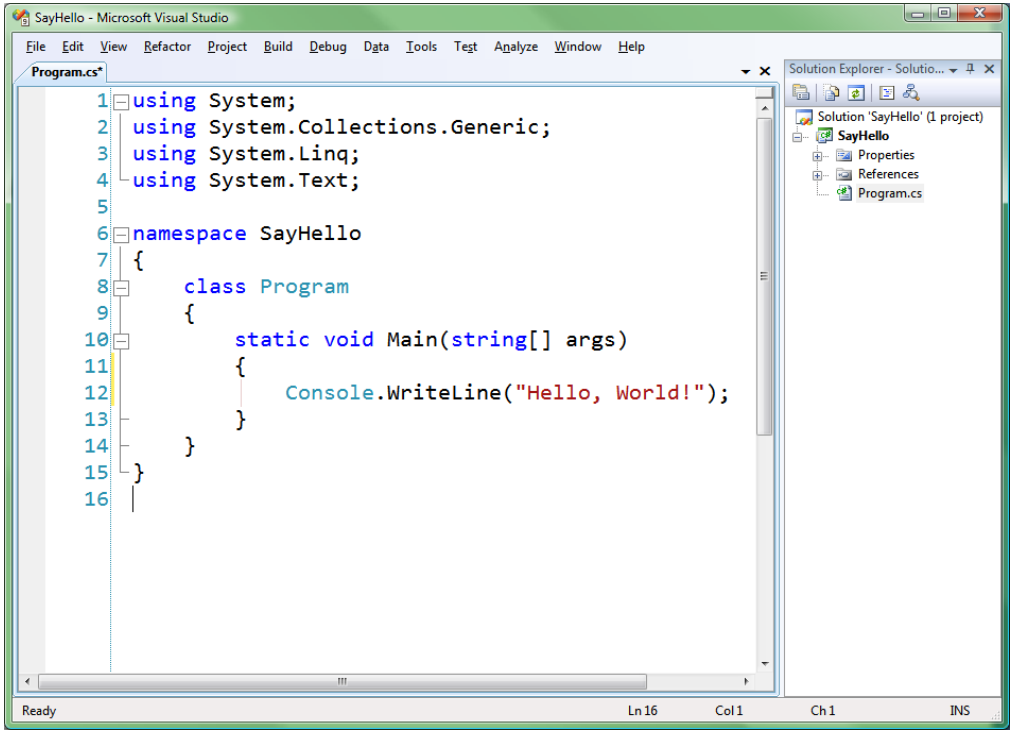
```
public static void Main()
{
    if (DateTime.Now.DayOfWeek == DayOfWeek.Monday)
    {
        Console.WriteLine("Another case of the Mondays!");
    }
}
```

csc.exe

- **The C# command line compiler**
  - Transforms C# code into Microsoft Intermediate Language
  - Produces an assembly (.dll, .exe)

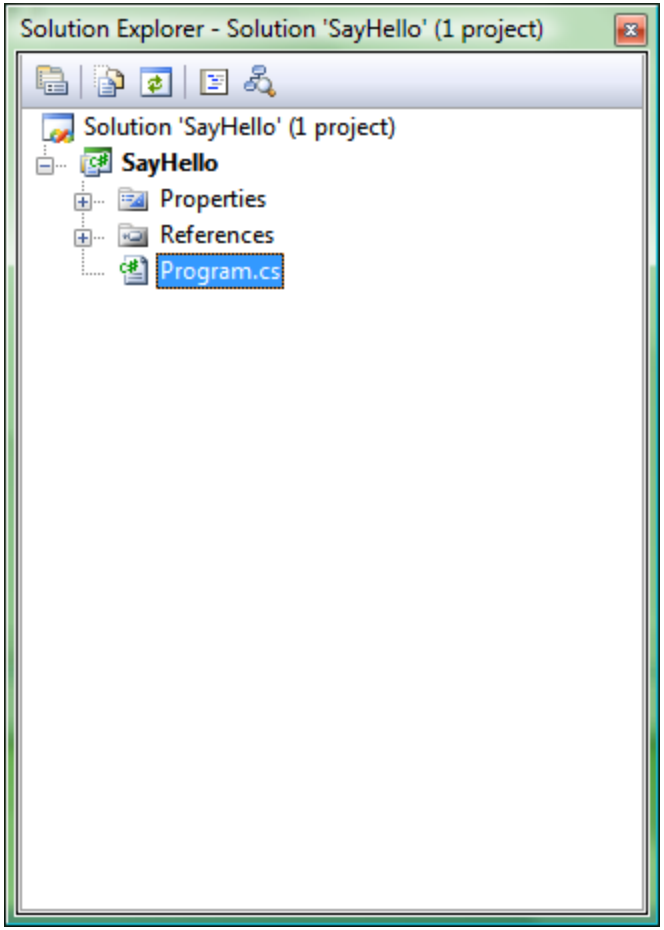
Visual Studio

- **An integrated development environment**
  - Edit C# (and other) files
  - Runs the C# compiler
  - Debugging
  - Testing



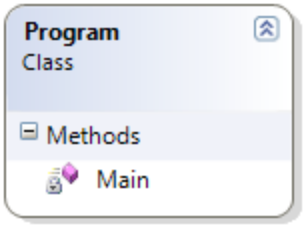
Solution Explorer

- **Will contain at least one project**
  - Contains one or more source code files
  - Each project produces an assembly
- **Projects organized under a solution**
  - Manage multiple applications or libraries



Types

- **C# is strongly typed**
  - One way to define a type is to write a class
  - Every object you work with has a specific type
  - 1,000s of types are built into the .NET framework
  - You can define your own custom types
- **Code you want to execute must live inside a type**
  - You can place the code inside a method
  - We'll explore other things you can add to a type later …



Primitive Types

Name	Description
Int32 (or int)	32 bit integer
Int64 (or long)	64 bit integer
Boolean (or bool)	true or false
Float (or float)	Single precision floating point
Double (or double)	Double precision floating point
Decimal (or decimal)	Fixed precision (financial)
DateTime	An instant in time (to 100 ns)
String (or string)	Text (as Unicode characters)

Namespaces

- **Namespace organize types**
  - Avoid type name collisions
  - Can define namespace in one or more places
- **Fully qualified type names**
  - Includes the assembly name
  - Includes the namespace
  - Includes the type name
- **using Directive**
  - Brings other namespaces into scope
  - No need to namespace qualify a Type

```
using System;
using System.Net;
using System.Data;
using System.Linq;
using System.Text;
```

Variables

- **Variables hold a value**
  - Variables always have a type
  - Must assign a value before you can use a variable
  - C# compiler will make sure types are compatible during assigment

```
static void Main(string[] args)
{
    int answer = 42;
    string name = "C#";

    name = answer;
}
```

```
(local variable) int answer

Error:
  Cannot implicitly convert type 'int' to 'string'
```

Statements & Expressions

- **A statement is an instruction**
  - A method is a series of statements
  - Statements end with semicolons
  - Statements are executed in the order they appear
- **Expressions are statements that produce a value**
  - Typically involve an operator (not required)
  - Can assign the expression value to a new variable, or test it

```
TakeOrder();
PackageOrder();
ShipOrder();
```

```
int x = 5;
int y = 10;
int answer = x + y;
```

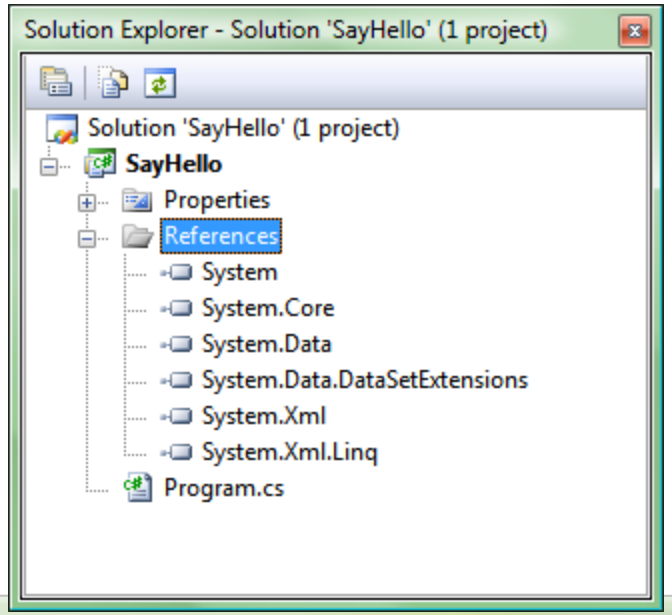
Operators

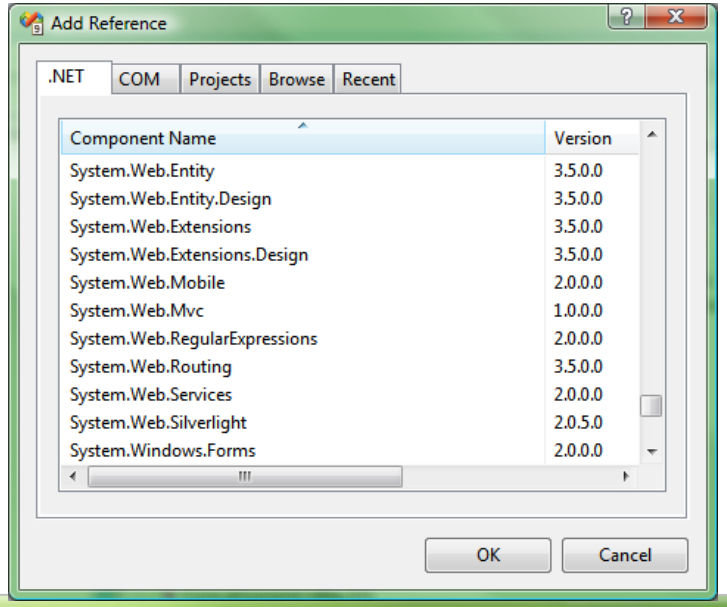
- **Specify an operation to perform on one or more variables**
  - Mathematical operators (+ , - , * , /)
  - Relational operators (< , > , <= , >=)
  - Equality operators (==, !=)
  - Conditional operators (&&, ||)
  - Assignment operators (=, +=, -=, *=, /=)

```
int x = 5;
int y = 10;
if (x != y)
{
    x++;
}
else
{
    y += x;
}
```

References

- **Allow you to use types in another assembly**
  - Object Browser is one way to examine types
  - Reference other assemblies in the FCL
  - Reference 3rd party assemblies
  - Reference other assemblies in solution





Summary

- **C# is one of the many languages for .NET**
  - Syntax similar to C++ and Java
  - Strongly typed
  - Statements and expressions
  - Operators
  - References