C# Fundamentals

Introduction to C#

CliZ Ware | LEARNING

Overview

.NET Framework

- .NET – a *software framework* (it's something your application will build on top of)

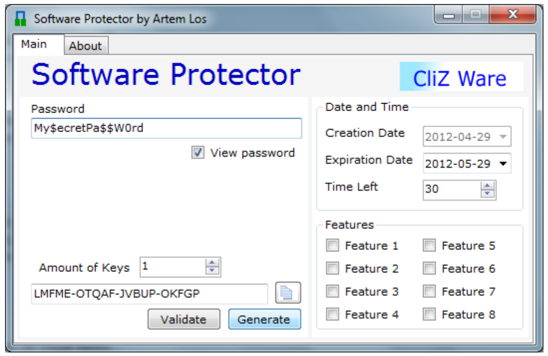
	Your Application	
Common Language Runtime (CLR)	Microsoft .NET	Framework Class Library (FCL)

.NET Framework

- It’s .NET Framework’s responsibility to provide all the services that different types of applications will need, for instance:
  - The ability to save information into a database
  - Read information from an XML file
  - Cryptography
  - Web applications
  - Numerical algorithms
  - Network communications

.NET Framework

- .NET – an *engineeric framework* because you can build wide kind of applications on top of it.
  - Web applications
  - Desktop applications
  - Windows services



Software Protector by Artem Los

Main | About

Software Protector

CliZ Ware

Password: My$ecretPa$$W0rd

☑ View password

Amount of Keys: 1

LMFME-OTQAF-JVBUP-OKFGP

Validate | Generate

Date and Time

Creation Date: 2012-04-29

Expiration Date: 2012-05-29

Time Left: 30

Features

☐ Feature 1	☐ Feature 5
☐ Feature 2	☐ Feature 6
☐ Feature 3	☐ Feature 7
☐ Feature 4	☐ Feature 8

.NET Framework

- Applications build on top of the .NET are platform independent, which means that you don’t need to care in what environment your application is being executed!

- Note, the .NET Framework needs to be installed on computer you want your software to execute.

CLR – Common Language Runtime

- It's an executive environment for your .NET applications. It brings the application alive, manages it, and when there is an error.

- When CLR manages you application, it provides these services.
  - Memory management
  - Security
  - Operating system
  - Language independence

FCL – Framework Class Library

- It's a library of functionality to build applications.
- It contains thousands of classes.
- It provides you with functionalities to do things such as:
  - Parse text
  - Use regular expression (RegEx)

FCL – Framework Class Library

1. One part of the class is Windows Presentation Foundation (WPF) that allows you to create applications that runs directly on user's desktop.

2. Another part of the class library is ASP.NET

3. Communications services

4. "System" namespace contains the commonly used once

CSC.exe

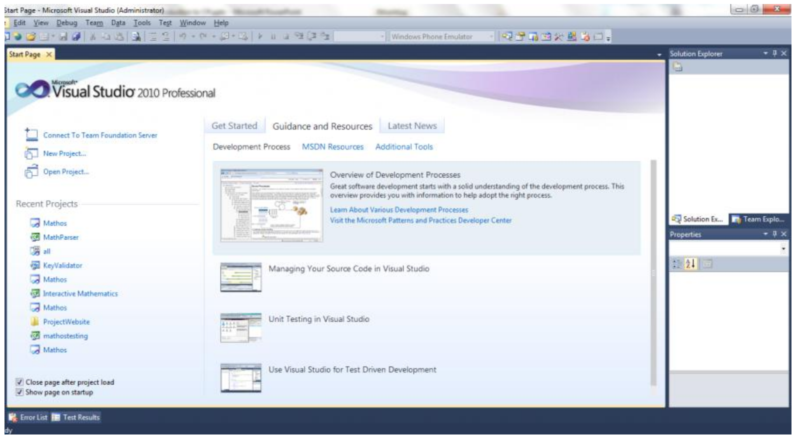
- The C# command line compiler
  - Transforms C# code into Microsoft Intermediate Language
  - Produces an assembly (.dll, .exe)
- These are the instructions to CLR, which are then converted to CPU instruction.
- CPU cannot execute .NET assembly

Visual Studio IDE

- An **I**ntegrated **D**evelopment **E**nvironment.

- The goal of an IDE is to put together all of the tools and functionalities that you need to build an applications into one place, and make it easy to get to the thing you need to do.

  - Edit C# (and other) files
  - Runs the C# compiler
  - Debugging
  - Testing

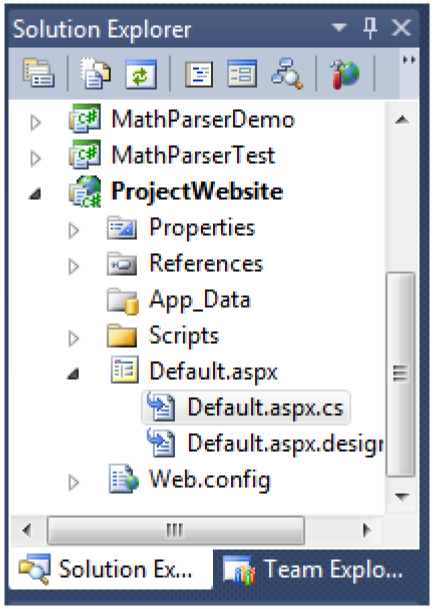

Start Page - Microsoft Visual Studio (Administrator)
Edit View Debug Team Data Tools Test Window Help
Windows Phone Emulator
Start Page
Solution Explorer
Microsoft Visual Studio 2010 Professional
Connect To Team Foundation Server
New Project...
Open Project...
Recent Projects
Mathos
MathParser
all
KeyValidator
Mathos
Interactive Mathematics
Mathos
ProjectWebsite
mathostesting
Mathos
Close page after project load
Show page on startup
Get Started
Guidance and Resources
Latest News
Development Process
MSDN Resources
Additional Tools
Learn About Various Development Processes
Visit the Microsoft Patterns and Practices Developer Center
Managing Your Source Code in Visual Studio
Unit Testing in Visual Studio
Use Visual Studio for Test Driven Development
Solution Ex...
Team Explo...
Properties
Error List
Test Results

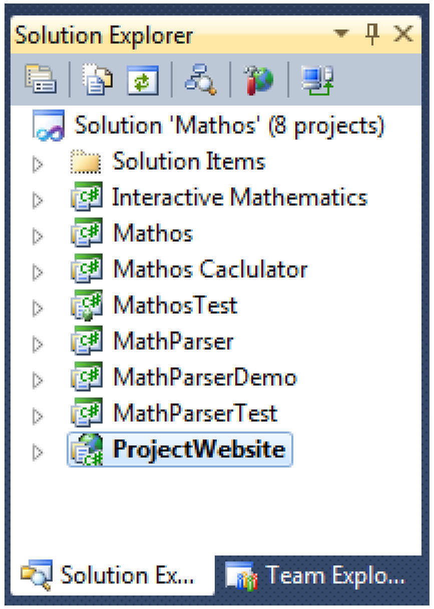
Overview of Development Processes

Great software development starts with a solid understanding of the development process. This overview provides you with information to help adopt the right process.

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

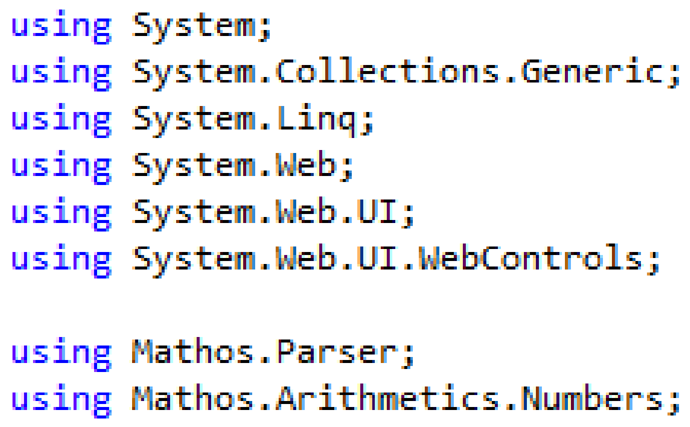
- C# is strongly typed
  - One way to define a type is to write a class
  - Every object you work with has a specific type
  - 1,000s of types are built into the .NET Framework
  - You can define your own custom types
- Code you want to execute must live inside a type
  - You can place the code inside a method

Primitive types

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 floating point (financial)
DateTime	An instant in time (to 100 ns)
String (or string)	Text (as Unicode characters)

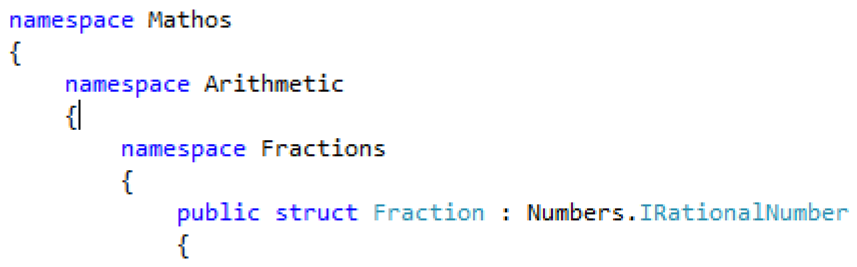
Namespaces

- 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.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

using Mathos.Parser;
using Mathos.Arithmetics.Numbers;
```



```
namespace Mathos
{
    namespace Arithmetic
    {|
        namespace Fractions
        {
            public struct Fraction : Numbers.IRationalNumber
            {
```

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 assignment

- Primitive types:

```
int age = 20;
bool IsAdult = true;
decimal salary = 3000.99M;
string welcomeMessage = "Hello World! :)";
```

- Custom types:

```
Fraction fractA = new Fraction(21, 7);
Fraction fractB = "21/7";
Fraction fractC = 3;
```

Statements & Expressions

- A statement is an instruction
  - A method is a series of statements
  - Statements end with semicolons
  - Statements are executed in 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

Reference

- 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 assembly
  - Reference other assemblies in the solution

Summary

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