A – Starting Classes

Code Samples – Documentation

# Examples – Overview

The following examples are used to illustrate this topic.

1. **Nothingness** - This class represents the absolute minimum code required to code a class. Even though it's not a very "useful" class, it does provide and introduction to the ideas of classes, keywords and identifiers.
2. **HelloWorld** - This adaptation of the classic "Hello World" program illustrates static methods. This example includes and introduces the concept of a "driver".
3. **Comments** - This class is not really "functional", but is intended to illustrate the basic types of comments allowed in the programming language.

### Nothingness

namespace Topic.A.Examples

{

public class Nothingness

{

}

}

### Salutation & HelloWorldDriver

namespace Topic.A.Examples

{

public class Salutation // Define a class called "Salutation"

{

public static string Greeting()

{

return "Hello World!";

} // end of Greeting()

public static string Greeting(string name)

{

return "Hello " + name;

} // end of Greeting(string)

public static string Farewell()

{

return "So long!";

} // end of Farewell()

} // end of Salutation class

}

namespace Topic.A.Examples

{

public class HelloWorld\_Driver

{

public static void Main(string[] args)

{

System.Console.WriteLine(Salutation.Greeting());

System.Console.WriteLine(Salutation.Greeting("Bob"));

System.Console.WriteLine(); // print a blank line

System.Console.WriteLine(Salutation.Farewell());

}

}

}

### Comments

/\*

\* File: Comments.cs

\* Author: Dan Gilleland

\* Date: 2010

\* Purpose: To illustrate multi-line, single-line, and XML comments.

\* This is a multi-line comment.

\*/

namespace Topic.A.Examples

{

/// <summary>

/// Comments illustrates multi-line, single-line, and XML comments.

/// </summary>

/// <remarks>

/// This class is a stand-alone class used to illustrate comments.

/// This particular XML comment is "attached" to the class as

/// a whole, while other XML comments are for fields or methods

/// in the class (a few methods and fields have been included for

/// illustration purposes).

/// [Author: Dan Gilleland]

/// </remarks>

public class Comments

{

/// <summary>

/// This is a method of <see cref="Comments"/>

/// </summary>

/// <returns>This method returns a string.</returns>

public static string SimpleMethod()

{

return "Hello World!"; // "Hello World!" is a literal string value

} // end of SimpleMethod()

/// <summary>

/// This is a one-sentence description of the method.

/// </summary>

/// <param name="name">This is where I describe the purpose of this parameter</param>

/// <returns>Here I describe what information is returned from this method.</returns>

/// <remarks>

/// This method has a single parameter.

/// </remarks>

public static int MethodWithParameter(string name)

{

return name.Length; // This is a single-line comment (must be end of physical line)

} // end of MethodWithParameter(string)

} // end of Comments class

}