

***C# BASICS***

**Training Assignments**

|  |  |
| --- | --- |
| Document Code | 25e-BM/HR/HDCV/FSOFT |
| Version | 1.1 |
| Effective Date | 20/11/2012 |

**Hanoi, 06/2019**

RECORD OF CHANGES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Effective Date | Change Description | Reason | Reviewer | Approver |
|  | 01/Oct/2018 | Create new | Draft |  |  |
|  | 01/Jun/2019 | Update template | Fsoft template | DieuNT1 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Contents

[Day 4: Assignment 14: Extension Method 4](#_Toc12288441)

[Objectives: 4](#_Toc12288442)

[Business needs: 4](#_Toc12288443)

[Prerequisites: 4](#_Toc12288444)

[Technologies: 4](#_Toc12288445)

[Function requirement: 4](#_Toc12288446)

[Code example: 4](#_Toc12288447)

[Use extension method: 5](#_Toc12288448)

|  |  |
| --- | --- |
|  | **CODE: Net.M.A014**  **TYPE: MEDIUM**  **LOC: 190**  **DURATION: 90 MINUTES** |

# Day 4: Assignment 14: Extension Method

**Objectives:**

* Understand and practice with Extension Method.
* Practice code in Visual Studio.
* Follow coding convention.

**Business needs:**

* TBD

**Prerequisites:**

* Working environment: Visual Studio 2013 or higher.Practice code in Visual Studio
* Each exercise is one project inside 1 solution.

**Technologies:**

The product implements one or more technology:

* C# Basic
* Extension method

**Function requirement:**

* The extension method application writes the ToStringFriendly method for the DateTime class of C #

**Code example:**

public static class DateTimeExtension

{

public static string ToStringFriendly(this DateTime dateTime)

{

TimeSpan span = (DateTime.Now - dateTime);

// Normalize time span

bool future = false;

if (span.TotalSeconds < 0)

{

// In the future

span = -span;

future = true;

}

// Test for Now

double totalSeconds = span.TotalSeconds;

if (totalSeconds < 0.9)

{

return "Now";

}

// Date/time near current date/time

string format = (future) ? "in {0} {1}" : "{0} {1} ago";

if (totalSeconds < 55)

{

// Seconds

int seconds = Math.Max(1, span.Seconds);

return String.Format(format, seconds,

(seconds == 1) ? "second" : "seconds");

}

if (totalSeconds < (55 \* 60))

{

// Minutes

int minutes = Math.Max(1, span.Minutes);

return String.Format(format, minutes,

(minutes == 1) ? "minute" : "minutes");

}

if (totalSeconds < (24 \* 60 \* 60))

{

// Hours

int hours = Math.Max(1, span.Hours);

return String.Format(format, hours,

(hours == 1) ? "hour" : "hours");

}

// Format both date and time

if (totalSeconds < (48 \* 60 \* 60))

{

// 1 Day

format = (future) ? "Tomorrow" : "Yesterday";

}

else if (totalSeconds < (3 \* 24 \* 60 \* 60))

{

// 2 Days

format = String.Format(format, 2, "days");

}

else

{

// Absolute date

if (dateTime.Year == DateTime.Now.Year)

format = dateTime.ToString(@"MMM d");

else

format = dateTime.ToString(@"MMM d, yyyy");

}

// Add time

return String.Format("{0} at {1:h:mm tt}", format, dateTime);

}

}

**Use extension method:**

DateTime dateTime = new DateTime(2019, 2, 20, 11, 30, 45, DateTimeKind.Local);

var friendlyString = dateTime.ToStringFriendly();

////return friendlyString: Yesterday at 11:30 AM