



คู่มือ CODING ด้วย

VISUAL BASIC

ฉบับผู้เริ่มต้น

2019

- ❖ ครอบคลุมการพัฒนาแอปพลิเคชันด้วย .NET 5
- ❖ อ่านเข้าใจง่าย อธิบายอย่างเป็นขั้นตอน มีตัวอย่างประกอบทุกหัวข้อ
- ❖ มีตัวอย่างระบบงานพื้นฐานเพื่อใช้ในการเรียนรู้และพัฒนาสู่ระบบงานจริง
- ❖ เหมาะสำหรับนักเรียน นักศึกษา และผู้ที่เริ่มต้นพัฒนาแอปพลิเคชัน



IDC

PREMIER

มีเพียง “ความรู้” เท่านั้นที่มนุษย์ใช้พลิก “โลก” และเปลี่ยน “ชีวิต”
เราจึงสร้างสรรค์ และส่งมอบ “ความรู้” ในรูปแบบที่ดีกว่า
เพื่อให้คนไทย “เรียนรู้” ได้ตลอดชีวิต



Think
Beyond



คู่มือ coding ด้วย Visual Basic 2019 ฉบับผู้เริ่มต้น

Author	ศุภชัย สมพานิช https://www.facebook.com/thaivb.net
Editorial	กิตตินันท์ พลสวัสดิ์ kitinan_p@idcpremier.com
Graphic Designers	ยุทธนา เกิดประดิษฐ์, สิริลักษณ์ วาระเลิศ
Page Layout	วรวิทย์ วรจินต์
Proofreader	พรพรรณรัตน์ ชูราษฎร์
Publishing Coordinators	วรพล ณิชกุล, สุพัตรา อาจปัฐ, ศรัณย์ คมขำ

Visual Studio 2019 เป็นเครื่องหมายการค้าของบริษัท Microsoft Corp. และเครื่องหมายการค้าอื่นๆ ที่อ้างถึงเป็นของบริษัทนั้นๆ สงวนลิขสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 โดยบริษัท ไอดีซี พรีเมียร์ จำกัด ห้ามลอกเลียนไม่ว่าส่วนใดส่วนหนึ่งของหนังสือเล่มนี้ไม่ว่าในรูปแบบใดๆ นอกจากจะได้รับอนุญาตเป็นลายลักษณ์อักษรจากผู้จัดพิมพ์เท่านั้น

บริษัท ไอดีซี พรีเมียร์ จำกัด จัดตั้งขึ้นเพื่อเผยแพร่ความรู้ที่มีคุณภาพสู่ผู้อ่านชาวไทย เรายินดีรับงานเขียนของนักวิชาการและนักเขียนทุกท่าน ท่านผู้สนใจกรุณาติดต่อผ่านทางอีเมลที่ infopress@idcpremier.com หรือทางโทรศัพท์หมายเลข 0-2962-1081 (อัตโนมัติ 10 คู่สาย) โทรสาร 0-2962-1084

PUBLISHED AND DISTRIBUTED BY



พิมพ์ครั้งที่ 1 กันยายน 2563

บริษัท ไอดีซี พรีเมียร์ จำกัด

200 หมู่ 4 ชั้น 19 ห้อง 1901 อาคารจัสมินอินเตอร์เนชั่นแนล
ทาวเวอร์ ถ.แจ้งวัฒนะ อ.ปากเกร็ด จ.นนทบุรี 11120
โทรศัพท์ 0-2962-1081 (อัตโนมัติ 10 คู่สาย)
โทรสาร 0-2962-1084

สมาชิกสัมพันธ์

โทรศัพท์ 0-2962-1081-3 ต่อ 121
โทรสาร 0-2962-1084

ร้านค้าและตัวแทนจำหน่าย

โทรศัพท์ 0-2962-1081-3 ต่อ 112-114
โทรสาร 0-2962-1084

ข้อมูลทางบรรณานุกรม

ศุภชัย สมพานิช
คู่มือ coding ด้วย Visual Basic 2019 ฉบับผู้เริ่มต้น
นนทบุรี : ไอดีซีฯ, 2563
448 หน้า
1. ไพธอน
2. ภาษาที่ใช้ในการเขียนโปรแกรม
I ชื่อเรื่อง
005.13
ISBN 885-916-100-801-9
ราคา 345 บาท

คำนำ

หนังสือเล่มนี้เป็นการนำเสนอเนื้อหาการเขียนโปรแกรมด้วยภาษา VB สำหรับคุณผู้อ่านที่ต้องการศึกษาการเขียนโปรแกรม ในปัจจุบันการเขียนโปรแกรมถือเป็นสกิลติดตัวอีกด้านหนึ่งที่ทุกสายอาชีพควรมีติดตัวไว้ ภาษา VB เป็นอีก 1 ภาษาที่น่าศึกษา

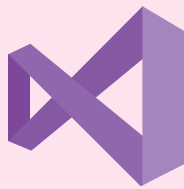
เนื้อหาที่นำเสนอในเล่มนี้ สามารถใช้งานครอบคลุมถึง .NET 5 เป็นทิศทางที่บริษัทไมโครซอฟท์กำหนดให้รวม .NET Core เวอร์ชัน 3.x เข้ากับ .NET Framework เวอร์ชัน 4.x กลายเป็น .NET 5 เพียงหนึ่งเดียว หมายความว่า เนื้อหาที่คุณศึกษา ยังคงใช้งานได้ต่อไป

สำหรับคุณผู้อ่านท่านใดที่ต้องการสอบถามปัญหา มีข้อเสนอแนะ หรือคำติตั้งข้อผิดพลาดต่างๆ สามารถติดต่อผู้เขียนได้ 4 ช่องทางคือ

1. แฟนเพจใน Facebook ที่ <https://www.facebook.com/thaivb.net>
2. กลุ่มใน Facebook ที่ <https://www.facebook.com/groups/thaivb.net>
3. ช่อง YouTube ที่ชื่อว่า : Thaivb.NET
4. แฟนเพจใน Blockdit ที่ <https://www.blockdit.com/thaivb.net>

ศุภชัย สมพานิช

มิถุนายน 2020



Visual Basic 2019

บทที่ 1	เตรียมความพร้อมก่อนเริ่มต้นเขียนโปรแกรมด้วยภาษา Visual Basic 2019.....	1
	ขอบเขตการนำเสนอของหนังสือเล่มนี้	1
	การดาวน์โหลดและติดตั้งโปรแกรม Visual Studio 2019.....	2
	ทำความรู้จักกับประเภทโปรเจกต์ที่น่าสนใจในขั้นต้น	3
	การอัปเดตโปรแกรม Visual Studio 2019.....	4
	การติดตั้งฟอนต์ Cascadia Code สำหรับเขียนโค้ดโดยเฉพาะ	7
	การสร้างโปรเจกต์ Windows Forms App (.NET Framework).....	8
	การออกแบบส่วนแสดงผลด้วยคอนโทรล (User Interface)	12
	การปรับแต่งคอนโทรลขั้นต้นด้วยวิธีแก้ไขคุณสมบัติ (Property).....	14
	การกำหนดเลขบรรทัด.....	16
	การทดสอบโปรเจกต์.....	17
บทที่ 2	พื้นฐานการทำงานกับโปรเจกต์ประเภท Windows Forms App (.NET Framework).....	19
	ทำความรู้จักกับฟอร์ม (Form)	19
	คุณสมบัติของ Form ที่สำคัญ.....	20
	การเพิ่มฟอร์มใหม่เข้ามาในโปรเจกต์.....	21
	วิธีการสั่งให้เปิดฟอร์มใหม่โดยการเขียนโค้ด	22
	หลักการเขียนโปรแกรมแบบ Event Driven Programming	24
	พื้นฐานการใช้ฟอร์มแบบ MDI และสร้างระบบเมนู (Menu).....	26
	การสร้างเมนูให้กับฟอร์มหลัก	28
	การลดขนาดโปรเจกต์ของ VB 2019.....	31
บทที่ 3	พื้นฐานการเขียนโปรแกรมด้วยภาษา Visual Basic	35
	การประกาศตัวแปรขึ้นมาใช้เก็บข้อมูล (Variable).....	35
	การใช้งานข้อมูลชนิดตัวเลข.....	36
	ความหมายของคำสั่ง Option Explicit On และ Option Strict On.....	37



กฎการตั้งชื่อตัวแปรในภาษา VB.....	38
คำสงวน (Reserved Words) ของภาษา VB	38
ขอบเขตการจัดเก็บข้อมูลแต่ละชนิดข้อมูลและขนาดหน่วยความจำที่ใช้.....	39
การค้นตัวเลขหลักพันด้วยเครื่องหมาย _	43
การกำหนดชนิดข้อมูลโดยอัตโนมัติ (Type Inference).....	43
การใช้งานเลขจำนวนเต็มชนิด BigInteger.....	46
ตัวดำเนินการด้านคณิตศาสตร์	48
ตัวดำเนินการด้านอัปเดตค่า	48
การเขียนหมายเหตุ (Comment) ในโค้ด	49
ขอบเขตของตัวแปร (Variable Scope)	49
การใช้งานตัวแปรระดับ Local และระดับฟอร์มในเวลาเดียวกัน	51
การใช้งานตัวแปรระดับ Block.....	54
การสร้างตัวแปรเก็บหลายค่าในเวลาเดียวกันด้วย ValueTuple	55
ค่าคงที่ (Constant).....	58

บทที่ 4 ทำงานกับข้อมูลชนิดข้อความ String และวันเวลา DateTime 61

พื้นฐานการทำงานกับข้อมูลชนิดข้อความ String.....	61
วิธีการตรวจสอบจำนวนตัวอักษรใน String	62
การต่อข้อความเข้าด้วยกัน.....	63
การแปลงตัวเลขที่อยู่ในฐานะข้อความ String ให้กลายเป็นข้อมูลชนิดตัวเลข	65
การแทนที่ข้อความด้วยเมธอด Replace().....	66
การแบ่งข้อความด้วยฟังก์ชัน Split()	66
การใช้งานข้อมูลชนิดตัวอักษร Char	67
การแปลงชนิดข้อความ String เป็นชนิดข้อมูลอาร์เรย์ของ Char	67
การจัดรูปแบบของตัวเลขทศนิยม	68
การแทรกค่าของตัวแปรเข้าไปในข้อความ String ด้วยเครื่องหมาย \$.....	71
ทำงานกับข้อมูลชนิดวันที่และเวลา (DateTime).....	72
การบวกเพิ่มวันที่กับข้อมูลประเภท DateTime.....	72

บทที่ 5 การตรวจสอบและควบคุม 77

พื้นฐานการตรวจสอบเงื่อนไขด้วยคำสั่ง If...Else.....	77
ตัวดำเนินการด้านเปรียบเทียบ	79

	การใช้ตัวดำเนินการ “และ” (AndAlso) และตัวดำเนินการ “หรือ” (OrElse)	80
	การตรวจสอบเงื่อนไขร่วมกับตัวดำเนินการด้านตรรกะ	80
	การตรวจสอบเงื่อนไขด้วยคำสั่ง If()	83
	การตรวจสอบเงื่อนไขด้วยคำสั่ง Select Case	84
	การวนลูปด้วยคำสั่ง For...Next	86
	การวนลูปด้วยคำสั่ง While...End While	88
	การวนลูปแบบ Do...Loop While	89
บทที่ 6	การใช้งานคอนโทรลเบื้องต้น	93
	ทำความรู้จักกับแถบคอนโทรล	93
	พื้นฐานการสร้างโปรแกรมรับข้อความจากผู้ใช้งานด้วยคอนโทรล TextBox	95
	คุณสมบัติของคอนโทรล TextBox ที่สำคัญ	98
	การตรวจสอบข้อมูลว่างในคอนโทรล TextBox	99
	การแสดงรูปภาพด้วยคอนโทรล PictureBox	101
	การสร้างส่วนแสดงผลแบบ list ด้วย ComboBox และ ListBox	103
	การเพิ่ม, ถอด และล้างรายการใน ListBox	108
	การสร้างส่วนแสดงผลแบบรายการด้วยคอนโทรล ListView	112
บทที่ 7	โครงสร้างข้อมูลแบบอาร์เรย์ (Array)	117
	พื้นฐานการใช้งานตัวแปรชุดแบบอาร์เรย์ (Array)	117
	การอ่านค่าตัวแปรอาร์เรย์ด้วยรูปแบบ For Each...Next	119
	การสร้างตัวแปรอาร์เรย์แบบ Type Inference	121
	การกลับด้านข้อมูลในอาร์เรย์	122
	การสร้างอาร์เรย์ด้วยคลาส ArrayList	124
	การถอดค่าซ้ำกันในอาร์เรย์	126
	การหาค่ามากที่สุดและน้อยสุดในอาร์เรย์	128
	การสร้างอาร์เรย์แบบหลายมิติ	132
บทที่ 8	การใช้งานฟังก์ชันและพารามิเตอร์	135
	รูปแบบฟังก์ชันในภาษา VB	135
	การสร้างฟังก์ชันที่มีชื่อเหมือนกันด้วยการทำ Overload Method	139
	ลักษณะของพารามิเตอร์ในฟังก์ชัน	143
	พารามิเตอร์ชนิดรับเข้า ByVal (Input Parameter)	143



	พารามิเตอร์ชนิดรับเข้าและส่งออกในเวลาเดียวกัน (ByRef)	144
	การกำหนดลักษณะเพิ่มเติมของพารามิเตอร์	145
	พารามิเตอร์แบบทางเลือก (Optional Parameter)	146
	การส่งค่าพารามิเตอร์แบบระบุชื่อ	148
	การส่งพารามิเตอร์แบบ Params	149
บทที่ 9	พื้นฐานการเขียนโปรแกรมเชิงวัตถุ (OOP Programming - OOP) ...	153
	การเขียนโปรแกรมเชิงวัตถุ (OOP) คืออะไร	153
	พื้นฐานการสร้างคลาสขึ้นมาใช้งานเอง	154
	ทำความรู้จักกับชนิดข้อมูลแบบ Anonymous Type	160
	การจัดประเภทคลาสด้วยระบบเนมสเปซ (Namespaces)	164
	วิธีการเรียกใช้เนมสเปซ	165
	แบบที่ 1 อาศัยคำสั่ง Imports	165
	แบบที่ 2 อาศัยการระบุชื่อเต็ม	166
	แบบที่ 3 อาศัยการตั้งชื่อ Alias	167
	แบบที่ 4 อาศัยคลาส Global	167
	ทำความรู้จักกับ Constructor, การระบุสมาชิกด้วยคำสั่ง Me และการกำหนดค่าเริ่มต้นด้วย Object Initializer	168
บทที่ 10	Inheritance & Polymorphism.....	175
	กฎพื้นฐาน 3 ข้อของ OOP	175
	พื้นฐานการสืบทอดคลาส (Inheritance)	176
	การสร้างคลาสแบบ Abstract Class (MustInherit)	180
	การใช้งาน Abstract Class ในฐานะ Dynamic Type	184
	สถานะที่หลากหลาย (Polymorphism)	186
บทที่ 11	การใช้งานคอนโทรลต่างๆ ที่น่าสนใจ	191
	แสดงข้อความด้วยคอนโทรล Label	191
	คุณสมบัติของคอนโทรล Label ที่สำคัญ	192
	การสร้างปุ่มกดด้วยคอนโทรล Button.....	192
	คุณสมบัติของคอนโทรล Button ที่สำคัญ	193
	การสร้างไฮเปอร์ลิงค์ด้วย LinkLabel.....	193
	คุณสมบัติของคอนโทรล LinkLabel ที่สำคัญ	196

การแสดงความคืบหน้าในการทำงานด้วยคอนโทรล ProgressBar	197
การแสดงคำอธิบายด้วย ToolTip	198
การแสดงผลแบบลำดับชั้นด้วยคอนโทรล TreeView	199
คุณสมบัติของคอนโทรล TreeView ที่สำคัญ.....	203
การสร้างตัวจับเวลาด้วย Timer.....	204
การสร้างตัวเลือกแบบเลือกได้ 1 อย่างด้วยคอนโทรล RadioButton	205
การสร้างตัวเลือกแบบหลายตัวเลือกด้วยคอนโทรล CheckBox	207
การจัดกลุ่มด้วยคอนโทรล GroupBox	209
การใช้งานตัวบรรจุ Panel	209
การสร้างส่วนแสดงผลแบบแท็บด้วยคอนโทรล TabControl	210
การสร้างตัวเลือกวันที่ด้วย DateTimePicker	211
คุณสมบัติของคอนโทรล DateTimePicker ที่สำคัญ	213
การสร้างปฏิทินด้วยคอนโทรล MonthCalendar	214
คุณสมบัติของคอนโทรล MonthCalendar ที่สำคัญ	216
การสร้างรายการแบบมีตัวเลือก CheckBox ด้วยคอนโทรล CheckedListBox.....	217
การกำหนดรูปแบบข้อมูลที่รับเข้ามาด้วย MaskedTextBox	219
การสร้างตัวเลือกแบบตัวเลขด้วย NumericUpDowns	221
ทำงานกับไฟล์ข้อความด้วยคอนโทรล RichTextBox, OpenFileDialog และ SaveFileDialog	223
คุณสมบัติของคอนโทรล RichTextBox ที่สำคัญ.....	223
คุณสมบัติของคอนโทรล OpenFileDialog ที่สำคัญ	223
คุณสมบัติของคอนโทรล SaveFileDialog ที่สำคัญ.....	224

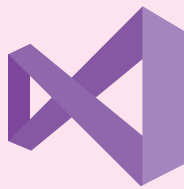
บทที่ 12 การเขียนโปรแกรมแบบ Generic 229

ทำความเข้าใจกับการเขียนโปรแกรมแบบ Generic ภายใน 30 นาที	229
ทำไม .NET จึงต้องมี Generic	234
พารามิเตอร์ T มีชนิดข้อมูลอะไรในขณะออกแบบ (Design-Time)	235
การสร้างคลาสแบบ Generic	236
การสร้างรายการ list แบบ Generic (Generic List)	238
การทำงานกับ Generic List ที่น่าสนใจ	240
ทำงานกับโครงสร้างข้อมูลแบบเข้าก่อน-ออกก่อน FIFO (First-In First-Out)	243
ทำงานกับโครงสร้างข้อมูลแบบเข้าหลัง-ออกก่อน LIFO (Last-In First-Out)	247



บทที่ 13	พื้นฐานการดักจับและตรวจสอบข้อผิดพลาด (Debug & Exception).....	251
	พื้นฐานการดักจับข้อผิดพลาดด้วย Try...Catch.....	251
	การดักจับข้อผิดพลาดกับโครงสร้างข้อมูล Array.....	254
บทที่ 14	พื้นฐานการใช้งานระบบฐานข้อมูล SQL Server 2019 Express Edition.....	257
	การดาวน์โหลดและติดตั้งฐานข้อมูล SQL Server 2019 Express Edition.....	257
	การทำงานกับระบบฐานข้อมูล SQL Server.....	262
	การดาวน์โหลดและติดตั้ง SQL Server Management Studio (SSMS).....	262
	พื้นฐานการใช้งานฐานข้อมูล SQL Server 2019 Express Edition ด้วย SSMS	264
	การยกเลิกป้องกันการแก้ไขโครงสร้างตาราง.....	269
	การสร้างความสัมพันธ์ระหว่างตารางในฐานข้อมูล SQL Server 2019.....	270
	การ Backup และ Restore ฐานข้อมูล SQL Server 2019	273
	การเพิ่มฐานข้อมูล Northwind เข้าไปใน SQL Server 2019 Express Edition	277
	การจัดการฐานข้อมูล SQL Server ด้วย Azure Data Studio (ADS)	279
	พื้นฐานการเชื่อมต่อกับฐานข้อมูล SQL Server ด้วย ADS.....	281
	การทำงานกับฐานข้อมูล SQL Server ในขั้นต้นด้วย ADS.....	283
บทที่ 15	พื้นฐานการทำงานกับฐานข้อมูล SQL Server ด้วย Entity Framework.....	285
	พื้นฐานการติดต่อฐานข้อมูล SQL Server ด้วย Entity Framework.....	286
	การเลือกดูข้อมูลบางฟิลด์โดยการทำ Anonymous Type.....	292
	วิธีการกำหนดโครงสร้างข้อมูลขึ้นมาใช้งานด้วย Anonymous Type	296
	การใช้งาน Anonymous Type ในฐานะแหล่งข้อมูลร่วมกับ Entity Framework	299
	พื้นฐานการค้นหาข้อมูลด้วย Entity Framework	303
	การแสดงผลข้อมูลร่วมกันตั้งแต่ 2 ตารางขึ้นไป.....	306
บทที่ 16	การจัดการข้อมูลในฐานข้อมูลด้วย Entity Framework.....	311
	พื้นฐานการจัดการข้อมูลโดยอาศัย Entity Framework.....	311
	การเพิ่มและลบข้อมูลแบบ 1 to Many (Master-Details).....	322
	การทำแบบฟอร์มป้อนข้อมูล (Data Entry) โดยอาศัย Entity Framework.....	328

บทที่ 17	พื้นฐานการพัฒนา Web Application ด้วย ASP.NET MVC.....	345
	โครงสร้างพื้นฐานของหน้าเว็บเพจด้วยภาษา HTML5.....	345
	วิธีการเข้ารหัสหน้าเว็บเพจ ให้สามารถแสดงผลภาษาไทย.....	346
	พื้นฐานการสร้างโปรเจกต์แบบ ASP.NET MVC.....	347
	การทดสอบรันโปรเจกต์ ASP.NET MVC.....	350
	ทำความเข้าใจกับโครงสร้างโปรเจกต์ของ ASP.NET MVC.....	353
	ส่วนแสดงผลของโปรเจกต์ ASP.NET MVC.....	355
บทที่ 18	Controller และ Model.....	359
	การสร้างคอนโทรลเลอร์ใหม่ในโปรเจกต์ ASP.NET MVC	359
	การสร้าง Hyperlink ใน ASP.NET MVC.....	364
	การส่งค่าให้กับเมธอดแบบมีพารามิเตอร์.....	367
	ทำงานกับส่วนของข้อมูล (Model).....	369
บทที่ 19	การใช้งาน ASP.NET MVC กับระบบฐานข้อมูล SQL Server.....	375
	การใช้งาน EF ร่วมกับฐานข้อมูล SQL Server ใน ASP.NET MVC	375
	การแสดงผลข้อมูลจากฐานข้อมูล SQL Server ด้วย EF.....	379
	การค้นหาข้อมูล	385
บทที่ 20	การจัดการข้อมูล (CRUD) ข้อมูลในฐานข้อมูล SQL Server ด้วย ASP.NET MVC.....	389
	การสร้างโปรเจกต์ CRUD ใน ASP.NET MVC.....	389
	การทำงานของโปรเจกต์ CRUD ของ ASP.NET MVC	408
บทที่ 21	การสร้างบริการ Back End ด้วย ASP.NET Web API.....	423
	ASP.NET Web API คืออะไร	423



Visual Basic 2019

เตรียมความพร้อมก่อนเริ่มต้นเขียนโปรแกรม ด้วยภาษา Visual Basic 2019

โลกของการเขียนโปรแกรมในยุคของ .NET ไม่ได้มีแต่เพียงแค่พัฒนาโปรแกรมบน PC เพียงอย่างเดียวเท่านั้น แต่ยังรวมไปถึงมือถือและแท็บเล็ตอีกด้วย ไมโครซอฟท์พัฒนา Visual Studio 2019 (เรียกสั้นๆ ว่า VS 2019) ขึ้นมา เพื่อรองรับการพัฒนาโปรแกรม 3 ช่องทาง คือ

1. พัฒนาโปรแกรมสำหรับ PC (หรือ Notebook) อยู่ในความรับผิดชอบของโปรเจกต์ประเภท Windows Forms Application ผลลัพธ์ของโปรเจกต์ประเภทนี้ คือ ไฟล์นามสกุล .exe หรือ .msi ที่เราค้นเคยกันเป็นอย่างดี
2. พัฒนาโปรแกรมบนเว็บ (ไม่ขึ้นกับแพลตฟอร์ม) อยู่ในความรับผิดชอบของโปรเจกต์ประเภท ASP.NET MVC
3. พัฒนาโปรแกรมสำหรับมือถือและแท็บเล็ต นั่นคือ Android/iOS Apps อยู่ในความรับผิดชอบของโปรเจกต์ประเภท Xamarin

ขอบเขตการนำเสนอของหนังสือเล่มนี้

เนื้อหาที่จะนำเสนอในหนังสือเล่มนี้ ครอบคลุมการพัฒนาโปรแกรมบน Desktop, Web, มือถือ และแท็บเล็ต โดยอาศัยภาษา Visual Basic 2019 (เรียกสั้นๆ ว่า VB 2019) ในระดับเบื้องต้นเท่านั้น แต่ก็มีเนื้อหาสาระเพียงพอที่จะต่อยอดในลำดับต่อไป ประกอบด้วยเนื้อหา 3 ส่วนใหญ่ๆ คือ

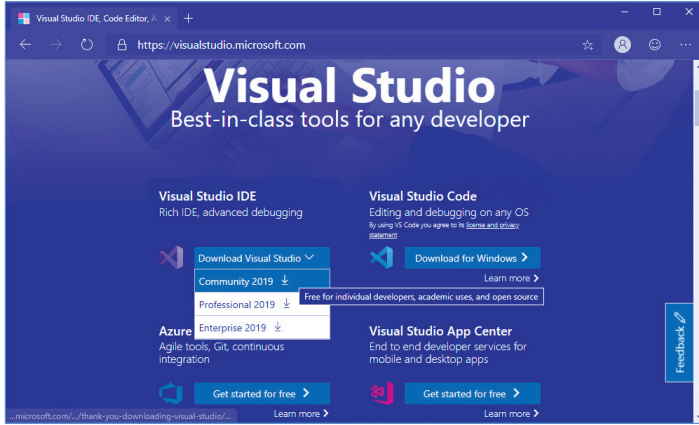
1. ศึกษาไวยากรณ์ภาษา VB และพัฒนาแอปสำหรับ Desktop โดยอาศัยโปรเจกต์ประเภท Windows Forms Application
2. พัฒนาแอปทำงานกับระบบฐานข้อมูล SQL Server ด้วย Entity Framework เรียกสั้นๆ ว่า EF
3. พัฒนา Web Application ด้วย ASP.NET MVC และยังรวมไปถึงแนะนำการพัฒนา Web API ในโลกของ .NET



การดาวน์โหลดและติดตั้งโปรแกรม Visual Studio 2019

การพัฒนาโปรแกรมในโลกของ .NET มีหลายวิธี หนังสือเล่มนี้อาศัยโปรแกรมที่ชื่อว่า Visual Studio 2019 โดยสามารถดาวน์โหลดมาใช้งานได้ฟรี มีขั้นตอนดังนี้

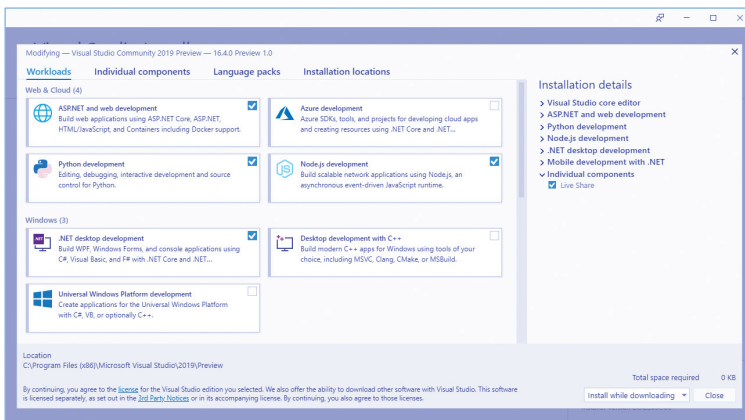
1. ไปที่เว็บไซต์ [www.visualstudio.com](https://visualstudio.com) เลือกดาวน์โหลดโปรแกรม Visual Studio 2019 เวอร์ชัน Community Edition เวอร์ชัน Preview มาใช้งานได้ฟรี ดังรูปที่ 1-1



รูปที่ 1-1 แสดงการดาวน์โหลดตัวติดตั้งโปรแกรม Visual Studio 2019

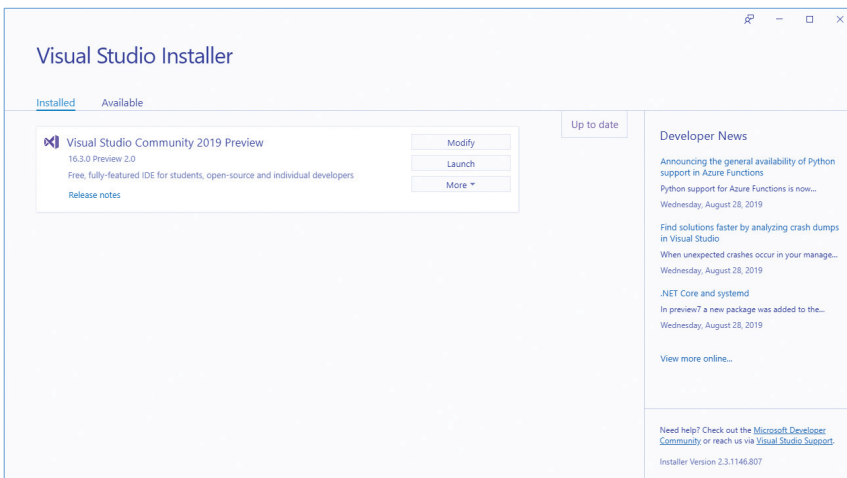
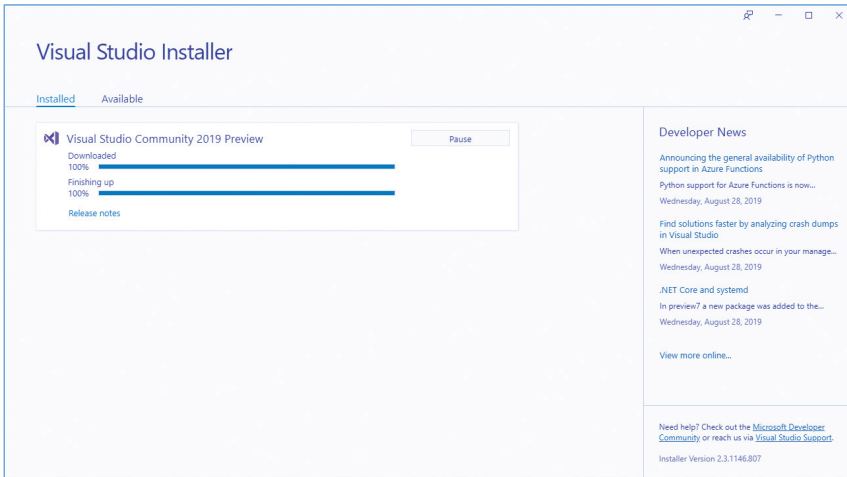
2. ดับเบิลคลิกไฟล์ที่ดาวน์โหลดมา เพื่อเข้าสู่ขั้นตอนการติดตั้งโปรแกรม Visual Studio 2019 สามารถเลือกประเภทโปรเจกต์ที่คุณสนใจได้อย่างอิสระ โดยขอบเขตการนำเสนอเนื้อหาของหนังสือเล่มนี้ ประกอบด้วย

- ASP.NET and web development สำหรับพัฒนาโปรเจกต์ประเภท Web Apps
- .NET desktop development สำหรับพัฒนาโปรเจกต์ที่รันบนระบบปฏิบัติการ Windows



รูปที่ 1-2 แสดงหน้าจอเลือกติดตั้งประเภทโปรเจกต์

3. ท้ายที่สุด ให้คุณรอการดาวน์โหลดและติดตั้งรายการต่างๆ ตามที่เลือกจนเสร็จสมบูรณ์
ถือว่าโปรแกรม Visual Studio 2019 พร้อมใช้งานแล้ว ดังรูปที่ 1-3



รูปที่ 1-3 แสดงการติดตั้งโปรแกรม Visual Studio เสร็จสมบูรณ์

ทำความรู้จักกับประเภทโปรเจกต์ที่น่าสนใจในขั้นต้น

โปรแกรม Visual Studio รองรับการพัฒนาแอปในแพลตฟอร์มต่างๆ มากมาย แยกติดตั้งแต่ละประเภทอย่างชัดเจน ที่น่าสนใจในขั้นต้นมีดังนี้

1. **ASP.NET and web development** การพัฒนา Web Apps หรือสร้างเว็บไซต์ในโลกของ .NET/.NET Core อยู่ในความรับผิดชอบของโปรเจกต์ประเภท ASP.NET มีโปรเจกต์ที่น่าสนใจ คือ ASP.NET Core MVC และ ASP.NET Core Web API

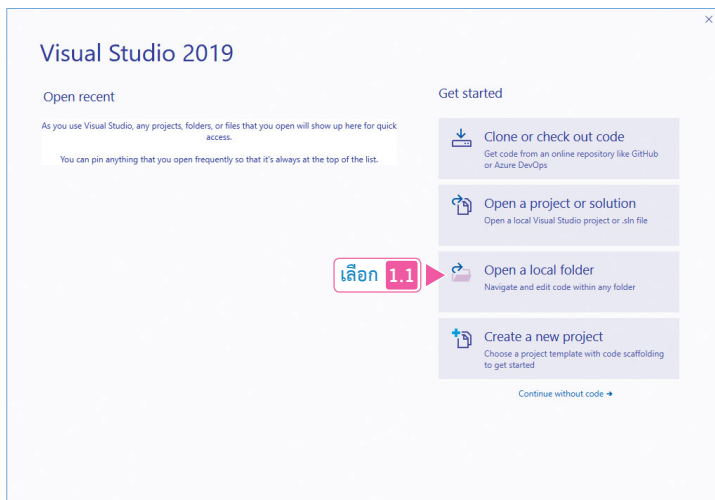


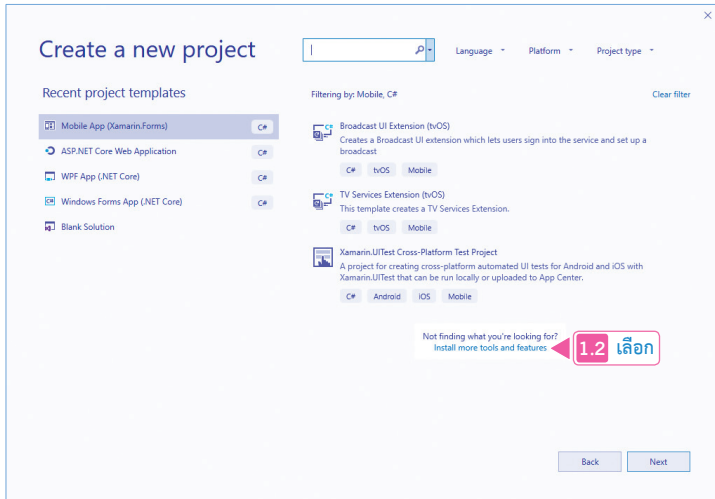
2. **Python development** คุณสามารถเขียนโปรแกรมภาษาไพธอนได้ด้วยเช่นกัน
3. **Azure development** เป็นโปรเจกต์ที่รองรับการใช้บริการต่างๆ บน Cloud ของไมโครซอฟท์ เรียกว่า Azure
4. **Node.js development** เป็นแพลตฟอร์มที่ใช้สำหรับสร้าง Web Apps หรือสร้างเว็บไซต์โดยอาศัยภาษา JavaScript ทั้งฝั่ง Client และฝั่ง Server เพียงภาษาเดียว เรียกว่า MEAN Stack
5. **.NET desktop development** เป็นโปรเจกต์ที่ใช้สร้างแอปที่ติดตั้งบน Windows ที่เราค้นเคยกันเป็นอย่างดี คือ ไฟล์นามสกุล .exe หรือ .msi โดยอาศัยภาษา VB หรือ C# นั่นเอง
6. **Universal Windows Platform development** เป็นโปรเจกต์ที่ใช้สำหรับสร้างแอปที่ติดตั้งบน Windows 10
7. **Desktop development with C++** เป็นโปรเจกต์ที่ใช้สร้างแอปที่ติดตั้งบน Windows เช่นกัน แต่ใช้ภาษา C++
8. **Mobile development with .NET** การพัฒนา Mobile Apps ในโลกของ .NET เป็นหน้าที่ของโปรเจกต์ที่เรียกว่า Xamarin สามารถพัฒนาได้ทั้ง iOS/Android Apps

การอัปเดตโปรแกรม Visual Studio 2019

ในปัจจุบันไมโครซอฟท์มักจะมีการอัปเดตโปรแกรม Visual Studio อยู่เสมอ เพื่ออัปเดตฟีเจอร์ใหม่ต่างๆ และยังรวมไปถึงการแก้ไขข้อผิดพลาดต่างๆ ที่เกิดขึ้นอีกด้วย โดยมีขั้นตอนดังนี้

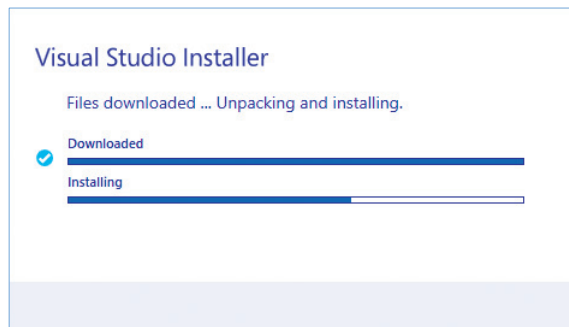
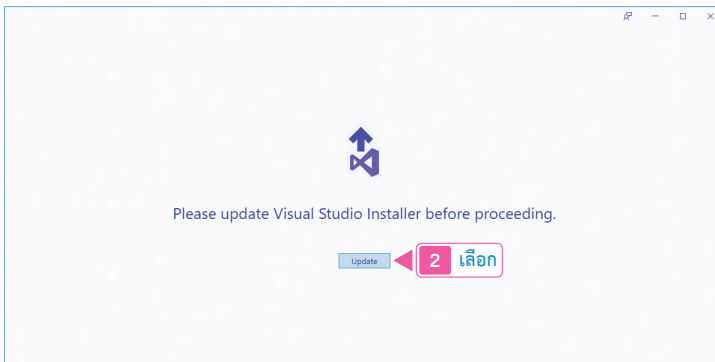
1. ที่หน้าจอเริ่มต้น เลือก Create a new project > Install more tools and features เพื่อเข้าสู่ขั้นตอนการอัปเดตโปรแกรม Visual Studio ดังรูปที่ 1-4





รูปที่ 1-4 แสดงการเข้าสู่ขั้นตอนการอัปเดตโปรแกรม Visual Studio

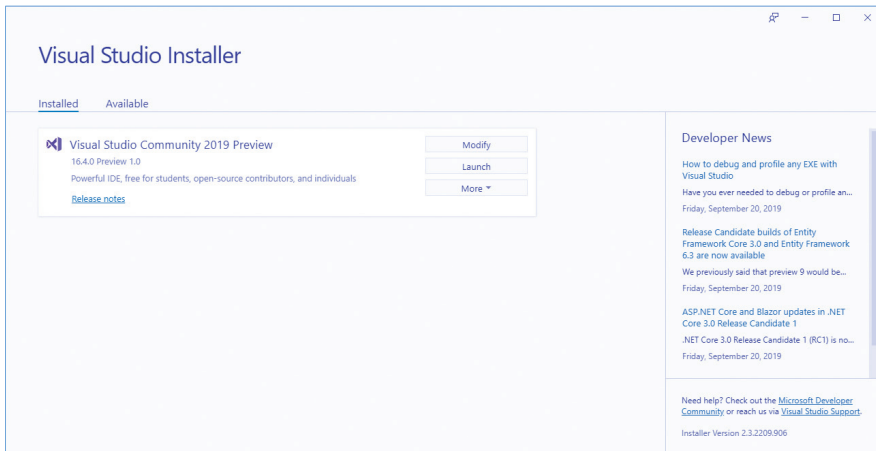
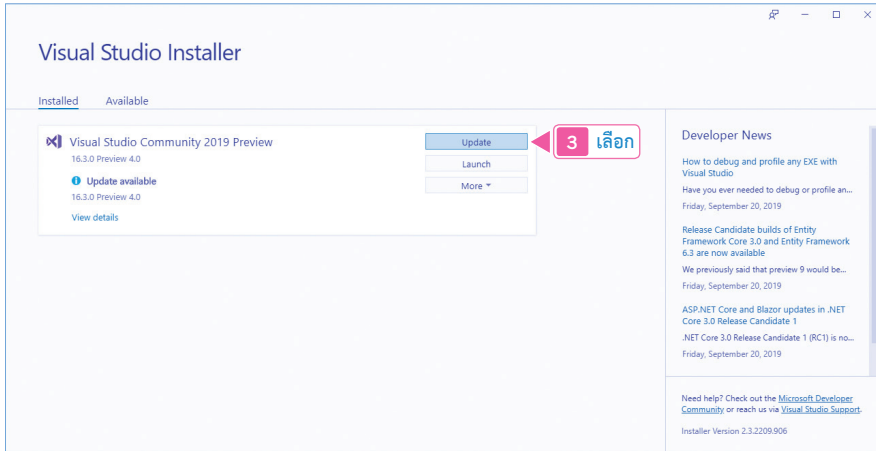
2. คลิกปุ่ม Update การอัปเดตโปรแกรม Visual Studio เป็นไปตามรายการที่คุณเลือกติดตั้งโดยอัตโนมัติทั้งหมด ดังรูป 1-5



รูปที่ 1-5 แสดงขั้นตอนเริ่มต้นอัปเดต



3. ท้ายที่สุด ให้คลิกที่ปุ่ม **Update** เพื่อเริ่มต้นดาวน์โหลดและติดตั้งรายการอัปเดตต่างๆ ตามเวอร์ชันปัจจุบันของคุณ ซึ่งเป็นขั้นตอนอัตโนมัติทั้งหมด ให้รอจนเสร็จสิ้นถือว่าจบขั้นตอนการอัปเดตโปรแกรม Visual Studio ดังรูปที่ 1-6

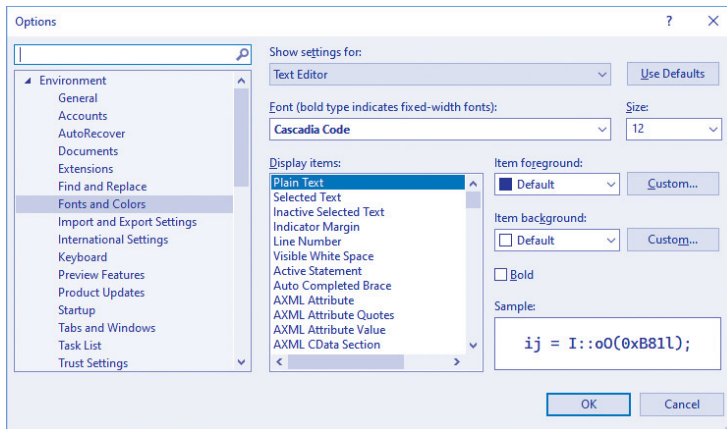
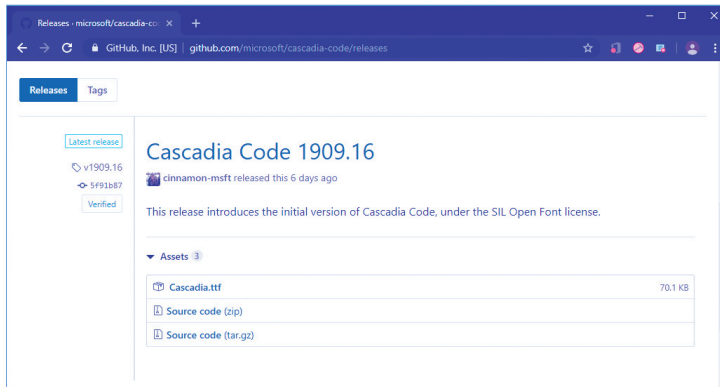


รูปที่ 1-6 แสดงการดาวน์โหลดและติดตั้งรายการอัปเดตโดยอัตโนมัติ

จากรูปที่ 1-6 คุณสามารถคลิกที่ลิงค์ **Release notes** เพื่อตรวจสอบว่า อัปเดตปัจจุบันมีรายการอะไรบ้าง

การติดตั้งฟอนต์ Cascadia Code สำหรับเขียนโค้ด โดยเฉพาะ

ไมโครซอฟท์สร้างฟอนต์สำหรับเขียนโค้ดโดยเฉพาะขึ้นมา ชื่อว่า Cascadia Code สามารถดาวน์โหลดมาใช้งานได้ที่เว็บไซต์ <https://github.com/microsoft/cascadia-code/releases> ดังรูปที่ 1-7



รูปที่ 1-7 แสดงเว็บไซต์ที่ให้ดาวน์โหลดฟอนต์เขียนโค้ด

จากรูปที่ 1-7 ให้คุณ Copy ฟอนต์ที่ได้มาไว้ในพาร C:\Windows\Fonts ก่อน จากนั้นเปิดโปรแกรม Visual Studio ขึ้นมา เลือกเมนู Tools > Options ที่หัวข้อ Environment > Fonts and Colors เลือกใช้ฟอนต์ที่ชื่อว่า **Cascadia Code**

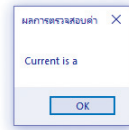
ข้อแตกต่างของฟอนต์ Cascadia Code กับฟอนต์อื่นๆ ที่น่าสนใจ คือ สามารถแสดงสัญลักษณ์พิเศษทางภาษาได้ เช่น เครื่องหมายหัวลูกศร เกิดมาจากการพิมพ์เครื่องหมาย = ติดกับเครื่องหมายมากกว่า > ดังรูปที่ 1-8



```

Form1.cs
ShortSwitch
20 private void Form1_Load(object sender, EventArgs e)
21 {
22     string txt = "a";
23     string result = CheckCondition(txt);
24     MessageBox.Show(result, "ผลการตรวจสอบค่า");
25 }
26
27 private string CheckCondition(string s) => s switch
28 {
29     "a" => "Current is a",
30     "b" => "Current is b",
31     _ => throw new ArgumentException()
32 };
33 }

```

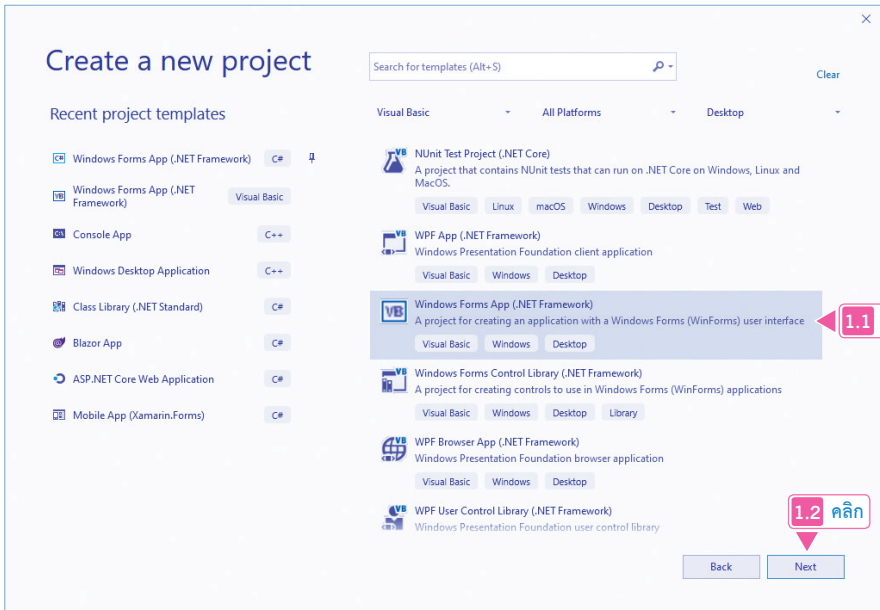


รูปที่ 1-8 แสดงเครื่องหมายลูกศรในโค้ด

การสร้างโปรเจกต์ Windows Forms App (.NET Framework)


โปรเจกต์แบบ Windows Forms App (.NET Framework) ถือเป็นโปรเจกต์ที่เหมาะสมกับการเริ่มต้นเขียนโปรแกรมมากที่สุด มีโครงสร้างโปรเจกต์ค่อนข้างง่าย มีขั้นตอนดังนี้

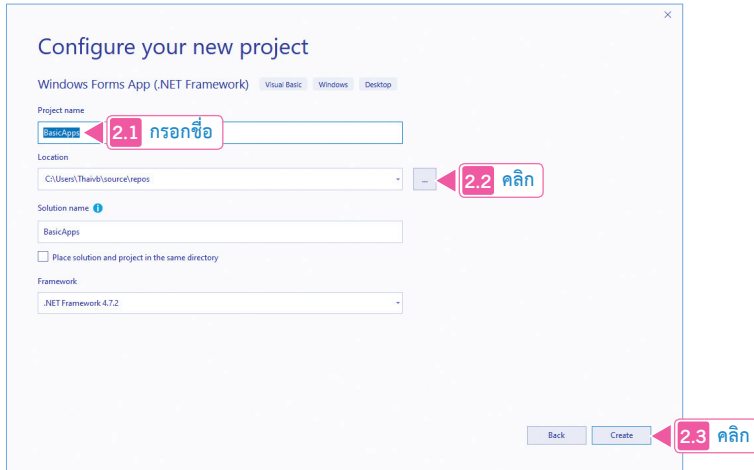
1. ที่หน้าจอเริ่มต้น ให้คุณเลือก Create a new project จากนั้นเลือกสร้างโปรเจกต์ประเภท Windows Forms App (.NET Framework) ดังรูปที่ 1-9



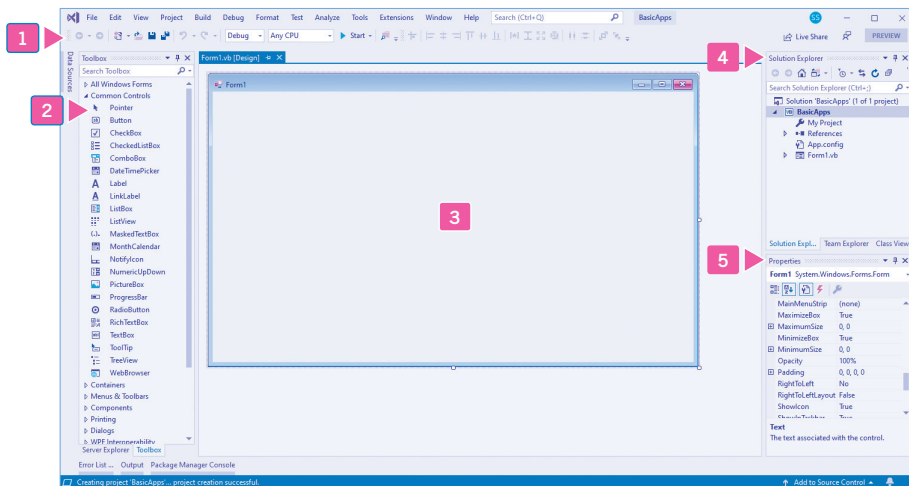
รูปที่ 1-9 แสดงการสร้างโปรเจกต์ Windows Forms App (.NET Framework) ของภาษา VB

2. เป็นการกำหนดรายละเอียดให้กับโปรเจกต์ที่จะสร้างขึ้นมา โดยที่

- **ช่อง Project name** หมายถึง ชื่อโปรเจกต์ ในกรณีนี้ตั้งชื่อว่า BasicApps
- **ช่อง Location** หมายถึง พาทที่ใช้จัดเก็บโปรเจกต์ปัจจุบัน โดยการคลิกที่ปุ่ม  เลือกโฟลเดอร์ตามที่คุณต้องการ ในกรณีนี้เก็บโปรเจกต์ไว้ที่พาท E:\Projects ดังรูปที่ 1-10



รูปที่ 1-10 แสดงการกำหนดรายละเอียดให้กับโปรเจกต์ที่จะสร้างขึ้นมา



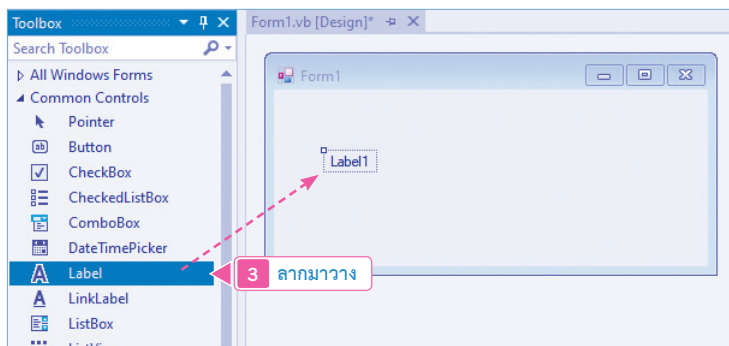
รูปที่ 1-11 แสดงสภาพแวดล้อมของ VS 2019 แบบ Windows Forms Application

จากรูปที่ 1-11 รายละเอียดแต่ละหน้าต่าง มีดังนี้

- **หมายเลข 1** แถบเครื่องมือสำหรับทดสอบโปรเจกต์ปัจจุบัน
- **หมายเลข 2** แถบคอนโทรลสำเร็จรูป ทำหน้าที่ออกแบบส่วนแสดงผล (User Interface เรียกว่า **UI**) ภายในโปรเจกต์ของคุณ

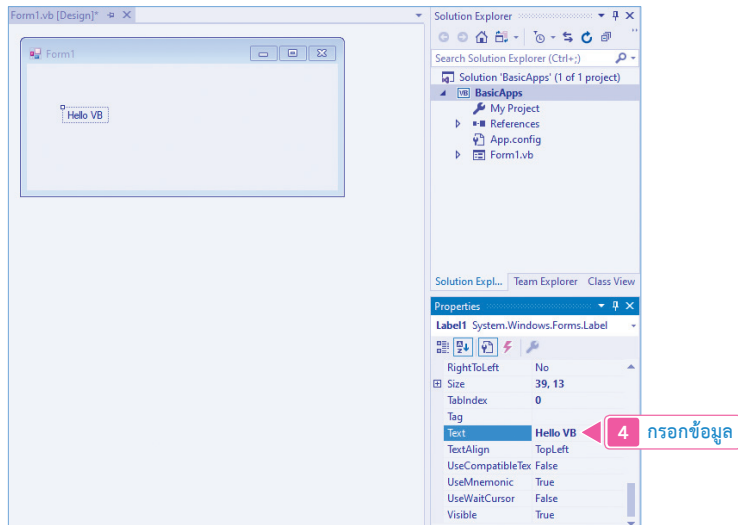


- **หมายเลข 3** ส่วนแสดงผลจำลอง คุณสามารถออกแบบส่วนแสดงผลให้กับหน้าจอต่างๆ ได้ที่นี่ โดยการลากคอนโทรลต่างๆ มาวางตามที่ต้องการ
 - **หมายเลข 4** เรียกว่า **หน้าต่าง Solution Explorer** ทำหน้าที่แสดงโครงสร้างโปรเจกต์ปัจจุบันของคุณว่าประกอบไปด้วยไฟล์อะไรบ้าง
 - **หมายเลข 5** เรียกว่า **หน้าต่างคุณสมบัติ (Properties)** ทำหน้าที่แสดงคุณสมบัติต่างๆ ของคอนโทรลที่กำลังถูกโฟกัสในส่วนแสดงผลจำลองว่ามีคุณสมบัติอะไรบ้าง โดยสามารถแก้ไขคุณสมบัติต่างๆ ได้ตามที่ต้องการเช่นกัน เช่น ตั้งชื่อคอนโทรลได้ที่คุณสมบัติ Name, กำหนดสีพื้นหลังได้ที่คุณสมบัติ BackColor เป็นต้น
3. ให้คุณเลือกใช้คอนโทรล Label เข้ามาทำหน้าที่แสดงข้อความ “Hello VB” โดยการลาก & วางใน Form1 ซึ่งทำหน้าที่เป็นส่วนแสดงผล ดังรูปที่ 1-12



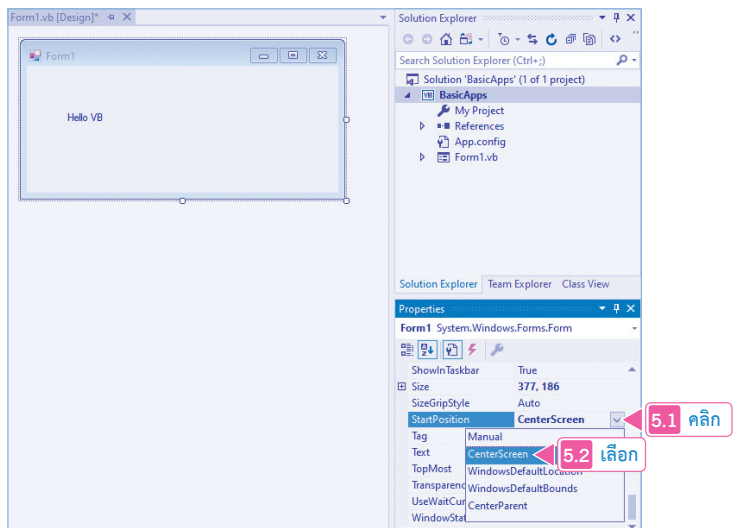
รูปที่ 1-12 แสดงการใช้งานคอนโทรล Label ใน Form1

4. การกำหนดข้อความใน Label ตัวนี้ ให้แก้ไขคุณสมบัติ **Text** ได้ที่หน้าต่าง Properties ให้เป็นข้อความตามที่คุณต้องการ



รูปที่ 1-13 แสดงการแก้ไขคุณสมบัติ Text ของคอนโทรล Label ตัวนี้

5. เป็นการกำหนดให้ Form1 ปรากฏอยู่กึ่งกลางหน้าจอและให้การโฟกัสที่ Form1 ก่อน โดยที่คุณสมบัติ **StartPosition** ทำหน้าที่กำหนดตำแหน่งของ Form1 ผู้เขียนเลือกแบบ **CenterScreen** หมายถึง ให้ปรากฏขึ้นมากึ่งกลางหน้าจอ ดังรูปที่ 1-14



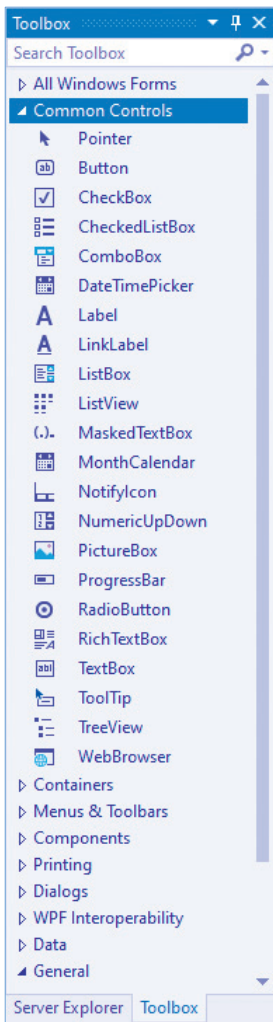
รูปที่ 1-14 แสดงการแก้ไขตำแหน่งของฟอร์มให้อยู่กึ่งกลางหน้าจอ

จากรูปที่ 1-14 ณ จุดนี้ ได้โปรเจกต์ตามที่เราต้องการแล้ว



การออกแบบส่วนแสดงผลด้วยคอนโทรล (User Interface)

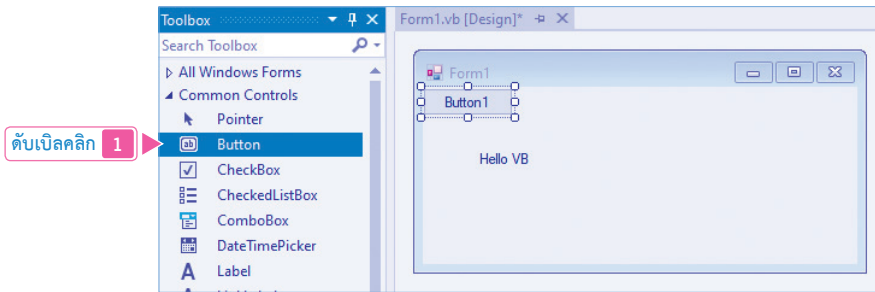
ไม่ว่าคุณจะทำโปรเจกต์ประเภทใดก็ตามใน Visual Studio 2019 ไมโครซอฟท์กำหนดให้สภาพแวดล้อมของตัวโปรแกรม Visual Studio 2019 มีลักษณะคล้ายกัน ใกล้เคียงกันมากที่สุดเท่าที่จะเป็นไปได้ เพื่อให้ นักพัฒนาคุ้นเคยกับตัวโปรแกรมได้ไม่ยาก



รูปที่ 1-15 แสดงแถบ Toolbox ของโปรเจกต์แบบ Windows Forms App (.NET Framework)

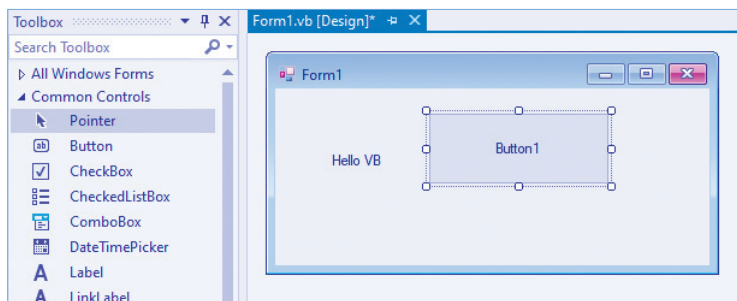
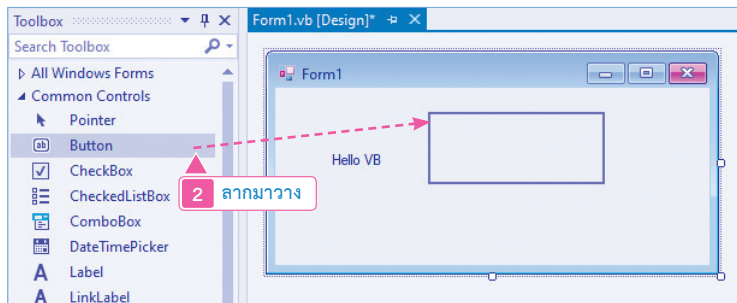
จากรูปที่ 1-15 แถบคอนโทรล (Control) อยู่ด้านซ้ายมือ ทำหน้าที่ออกแบบส่วนแสดงผล (User Interface เรียกว่า UI) ในขั้นตอนนี้คุณสามารถใช้งานคอนโทรลนี้ได้ 2 วิธี

1. ดับเบิลคลิกที่ตัวคอนโทรล เพื่อกำหนดให้ Visual Studio ใช้งานคอนโทรลดังกล่าวใน หน้าจอออกแบบโดยอัตโนมัติ เช่น ผู้เขียนต้องการใช้ปุ่มกด Button1 ดังรูปที่ 1-16



รูปที่ 1-16 แสดงการใช้ปุ่มกดด้วยวิธีการดับเบิลคลิกในแถบ Toolbox

2. คลิกเลือกคอนโทรลที่ต้องการใช้งาน แล้วนำไปวางในส่วนของฟอร์ม (Form) ดังรูปที่ 1-17



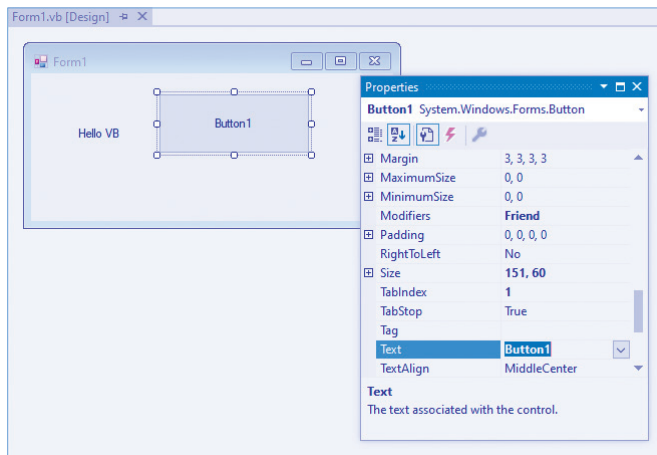
รูปที่ 1-17 แสดงการใช้ปุ่มกด Button ด้วยวิธีวาง

จากรูปที่ 1-17 เราได้ส่วนแสดงผลง่ายๆ มาแล้ว 1 หน้าจอ มีปุ่มกด Button เพียง 1 ปุ่ม



การปรับแต่งคอนโทรลขั้นต้นด้วยวิธีแก้ไขคุณสมบัติ (Property)

คอนโทรลต่างๆ ที่นำมาใช้งานสามารถปรับแต่งหรือแก้ไขได้ในหน้าต่างคุณสมบัติ (Properties) กล่าวคือ เมื่อคุณโฟกัสที่คอนโทรลตัวใดก็ตาม หน้าต่างคุณสมบัติจะเปลี่ยนไปโดยอัตโนมัติ



รูปที่ 1-18 แสดงรายการคุณสมบัติของปุ่มกด Button

จากรูปที่ 1-18 ส่วนแสดงผลปัจจุบันของผู้เขียนมีเพียง 1 ปุ่ม เมื่อถูกโฟกัสแล้ว หน้าต่างคุณสมบัติก็จะแสดงรายการคุณสมบัติต่างๆ ของคอนโทรล Button ให้เราโดยอัตโนมัติ

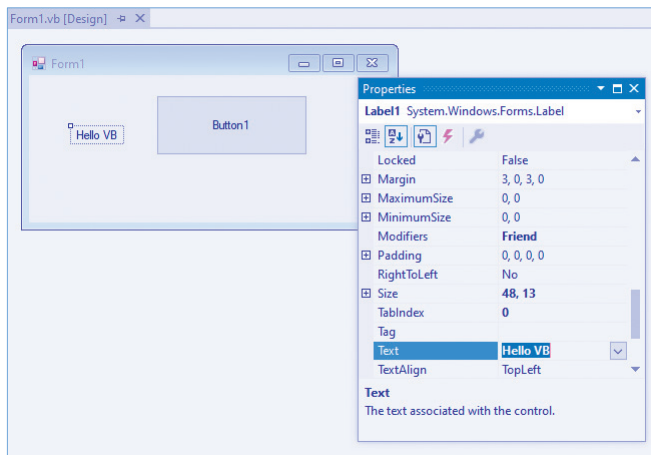
รายการคุณสมบัติของคอนโทรล Button ที่น่าสนใจ แสดงดังตารางต่อไปนี้

คุณสมบัติ	หน้าที่
Name	ทำหน้าที่ตั้งชื่อให้กับตัวคอนโทรล ใช้สำหรับอ้างอิงตอนเขียนโค้ด
Text	กำหนดข้อความที่จะแสดงในปุ่มกด
BackColor	กำหนดสีพื้นหลังให้กับปุ่มกด
ForeColor	กำหนดสีตัวอักษรที่อยู่ในปุ่มกด

Note

คุณสมบัติใดก็ตามที่ถูกแก้ไขจะแสดงด้วยตัวอักษรตัวหนา ส่วนตัวอักษรปกติ หมายถึง เป็นค่าเริ่มต้นที่โปรแกรม Visual Studio ตั้งไว้ โดยที่คุณไม่จำเป็นต้องไปแก้ไขให้ครบทุกคุณสมบัติ ให้แก้ไขเฉพาะเท่าที่จำเป็น

ต่อมาผู้เขียนลองใช้คอนโทรล Label ซึ่งทำหน้าที่แสดงข้อความ เมื่อคอนโทรล Label ได้รับโฟกัส พบว่าที่หน้าต่างคุณสมบัติก็จะแสดงรายการคุณสมบัติของคอนโทรล Label ให้เราโดยอัตโนมัติ



รูปที่ 1-19 แสดงรายการคุณสมบัติของโครงsa Label

รายการคุณสมบัติที่น่าสนใจ แสดงดังตารางต่อไปนี้

คุณสมบัติ	หน้าที่
Name	ทำหน้าที่ตั้งชื่อให้กับตัวคอนโทรล ใช้สำหรับอ้างอิงตอนเขียนโค้ด
Text	กำหนดข้อความที่จะแสดงใน Label
BackColor	กำหนดสีพื้นหลังให้กับคอนโทรล Label
ForeColor	กำหนดสีตัวอักษรที่อยู่ในคอนโทรล Label
AutoSize	กำหนดขนาดของคอนโทรล Label มี 2 แบบ คือ <ul style="list-style-type: none"> • True หมายถึง กำหนดให้ Label มีขนาดเท่าที่ข้อความแสดงอยู่ • False หมายถึง คุณสามารถกำหนดขนาดของ Label ได้อย่างอิสระ

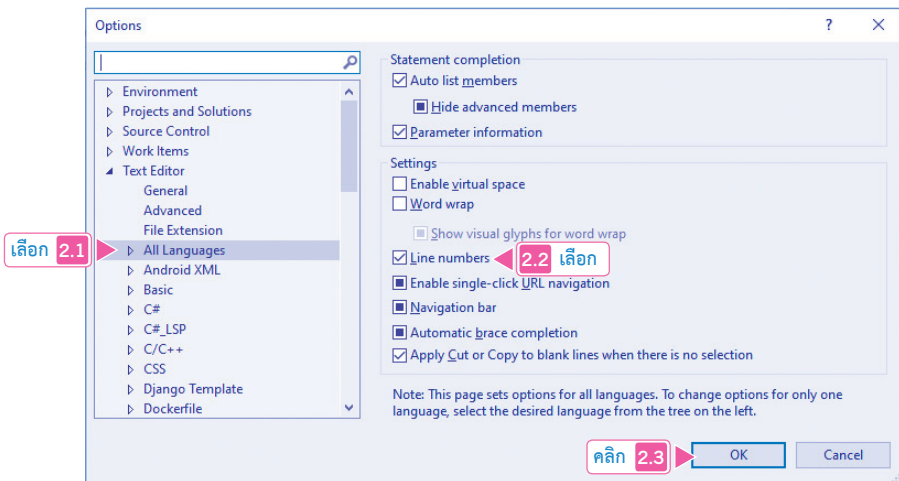
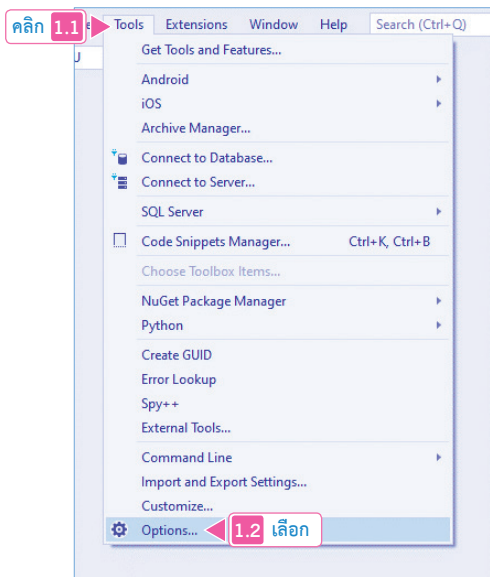
คอนโทรลแต่ละตัวก็จะมีรายการคุณสมบัติทั้งที่เหมือนกัน และต่างกันไปตามหน้าที่ของมันเอง



การกำหนดเลขบรรทัด

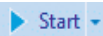
ส่วนการกำหนดให้ VS 2019 แสดงหมายเลขกำกับโค้ดในแต่ละบรรทัดทำได้ดังนี้

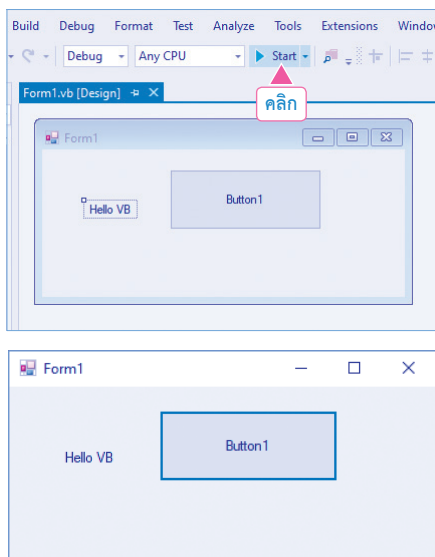
1. คลิกที่เมนู Tool > Options...
2. จะปรากฏหน้าต่าง Options ให้เลือกหัวข้อ Text Editor > All Languages เลือกตัวเลือก Line numbers ดังรูปที่ 1-20



รูปที่ 1-20 แสดงการกำหนดให้แสดงหมายเลขกำกับโค้ดแต่ละบรรทัด

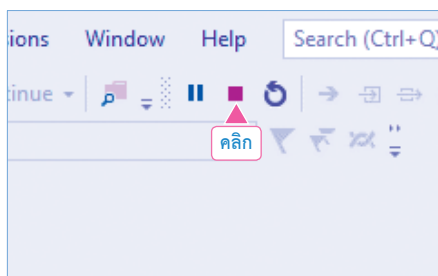
การทดสอบโปรเจกต์

หลังจากที่คุณออกแบบและเขียนโค้ดแล้ว ก็จะเข้าสู่ขั้นตอนการทดสอบโปรเจกต์ว่ามีการทำงานหรือมีผลการทำงานเป็นไปตามที่เราต้องการหรือไม่ โดยการคลิกที่ปุ่ม  เพื่อเริ่มต้นทดสอบรันโปรเจกต์



รูปที่ 1-21 แสดงการเริ่มทดสอบโปรเจกต์

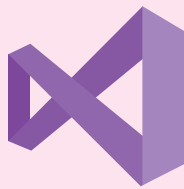
ในกรณีที่คุณต้องการหยุดรันโปรเจกต์ปัจจุบัน ให้คุณคลิกที่ปุ่ม  ดังรูปที่ 1-22



รูปที่ 1-22 กรณีต้องการหยุดรันโปรเจกต์ปัจจุบัน

สรุปท้ายบท

เนื้อหาในบทนี้เป็นเพียงขั้นตอนการเตรียมสภาพแวดล้อมให้เครื่องของคุณพร้อมใช้งานเท่านั้น เนื้อหาในบทต่อไปจะเข้าสู่โลกของการเขียนโปรแกรมอย่างแท้จริง



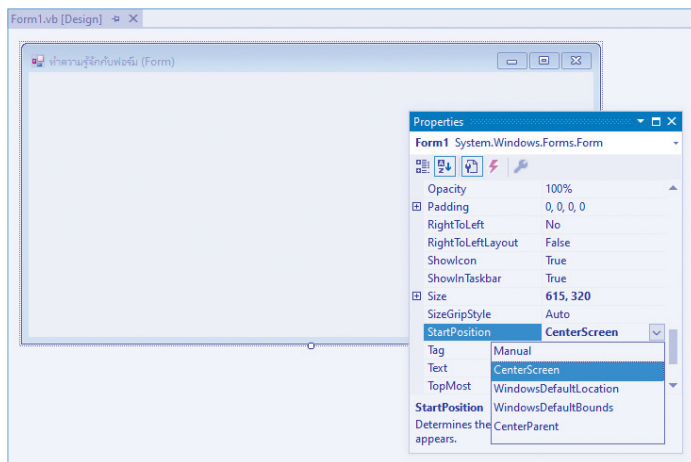
Visual Basic 2019

พื้นฐานการทำงานกับโปรเจกต์ประเภท Windows Forms App (.NET Framework)

ฟอร์ม (Form) ทำหน้าที่เหมือนกับเป็นตัวบรรจุ (Container) คอนโทรลต่างๆ เพื่อสร้างส่วนแสดงผลตามที่คุณออกแบบไว้ เนื้อหาในบทนี้เราจะมาดูกันว่ามื่ออะไรสนใจในขั้นต้นบ้าง

ทำความเข้าใจกับฟอร์ม (Form)

เมื่อคุณสร้างโปรเจกต์แบบ Windows Forms Application ก็จะได้ฟอร์ม (Form) ที่มีชื่อว่า Form1 เป็นหน้าจอเริ่มต้น เรามักจะกำหนดให้แบบฟอร์มอยู่กึ่งกลางหน้าจอ (หรือกึ่งกลางส่วนแสดงผลเสมอ) โดยการกำหนดคุณสมบัติ **StartPosition** เท่ากับ CenterScreen ดังรูปที่ 2-1



รูปที่ 2-1 แสดงรายการคุณสมบัติของฟอร์ม

จากรูปที่ 2-1 เมื่อคุณโฟกัสที่ Form1 หน้าต่างคุณสมบัติก็จะแสดงรายการคุณสมบัติของ Form1 เช่นกัน



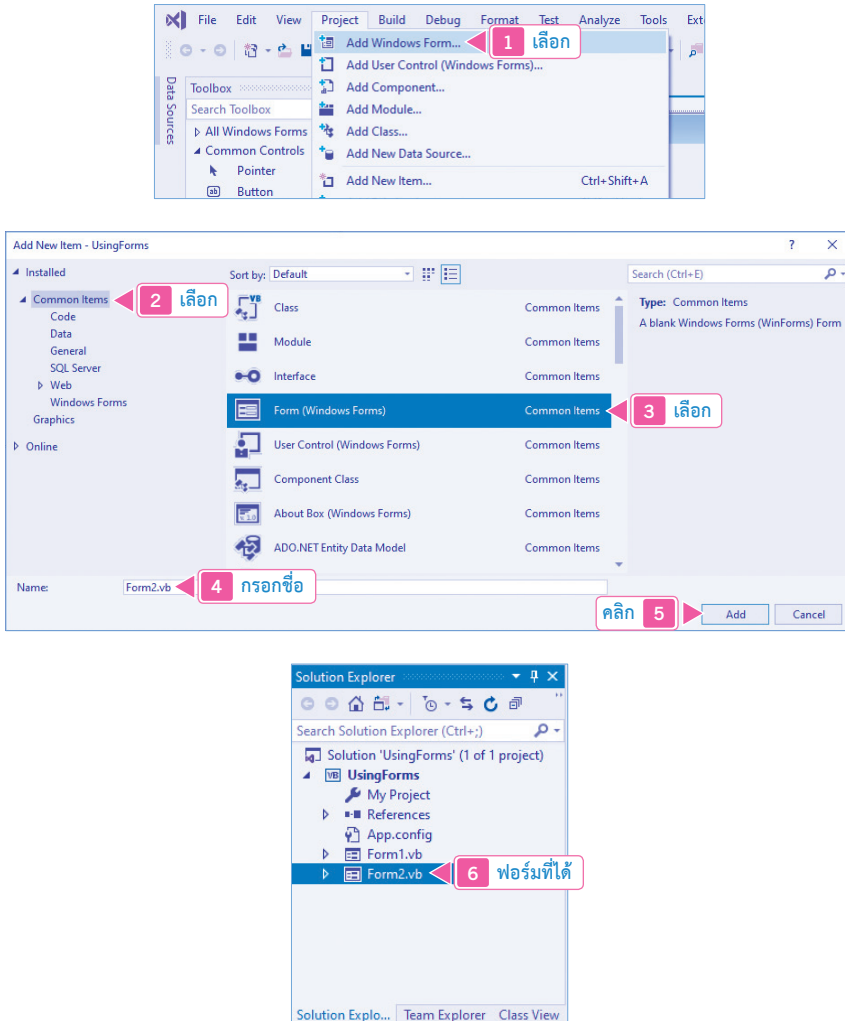
คุณสมบัติของ Form ที่สำคัญ

แสดงดังตารางต่อไปนี้

คุณสมบัติ	หน้าที่
Name	ทำหน้าที่ตั้งชื่อให้กับฟอร์ม ใช้สำหรับอ้างอิงตอนเขียนโค้ด
Text	กำหนดข้อความที่จะแสดงในแถบ Title Bar
BackColor	กำหนดสีพื้นหลังให้กับฟอร์ม
ForeColor	กำหนดสีตัวอักษรที่อยู่ในฟอร์ม
FormBorderStyle	กำหนดรูปแบบเส้นขอบของฟอร์ม โดยที่ <ul style="list-style-type: none"> • None หมายถึง ไม่มีแถบไตเติลบาร์ • FixedSingle หมายถึง ปรับขนาดฟอร์มไม่ได้ แต่มีปุ่ม Minimize, Maximize และปุ่ม Close • Fixed3D หมายถึง กำหนดให้มีเส้นขอบแบบ FixedSingle แต่มีแนวลิคด้วย • FixedDialog หมายถึง กำหนดให้เป็นแบบ FixedSingle แต่ใช้ในรูปแบบไดอะล็อกโต้ตอบกับผู้ใช้งาน • Sizable หมายถึง แบบฟอร์มปรับขนาดได้ • FixedToolWindow หมายถึง แบบฟอร์มปรับขนาดไม่ได้ และไม่มีปุ่ม Minimize และปุ่ม Maximize • SizableToolWindow หมายถึง แบบฟอร์มปรับขนาดได้ และไม่มีปุ่ม Minimize และปุ่ม Maximize
MaximizeBox	กำหนดให้แบบฟอร์มแสดงปุ่มขยายขนาดฟอร์ม โดยที่ <ul style="list-style-type: none"> • True หมายถึง ให้มีปุ่ม Maximize มุมขวา-บน • False หมายถึง ซ่อนปุ่ม Maximize
MinimizeBox	กำหนดให้แบบฟอร์มแสดงปุ่มย่อฟอร์ม โดยที่ <ul style="list-style-type: none"> • True หมายถึง ให้มีปุ่ม Minimize มุมขวา-บน • False หมายถึง ซ่อนปุ่ม Minimize
IsMdiContainer	กำหนดให้ฟอร์มทำหน้าที่บรรจุฟอร์มอื่นๆ แบบหลายหน้าต่างได้หรือไม่ โดยที่ <ul style="list-style-type: none"> • True หมายถึง กำหนดให้ฟอร์มเป็นตัวบรรจุแบบ MDI • False หมายถึง กำหนดให้เป็นแบบฟอร์มปกติ
WindowState	กำหนดขนาดของฟอร์ม โดยที่ <ul style="list-style-type: none"> • Normal หมายถึง กำหนดให้ฟอร์มมีขนาดตามที่เรากดปุ่ม • Minimized หมายถึง กำหนดให้ฟอร์มมีขนาดย่อลงในแถบ Taskbar • Maximized หมายถึง กำหนดให้ฟอร์มมีขนาดเต็มหน้าจอส่วนแสดงผล

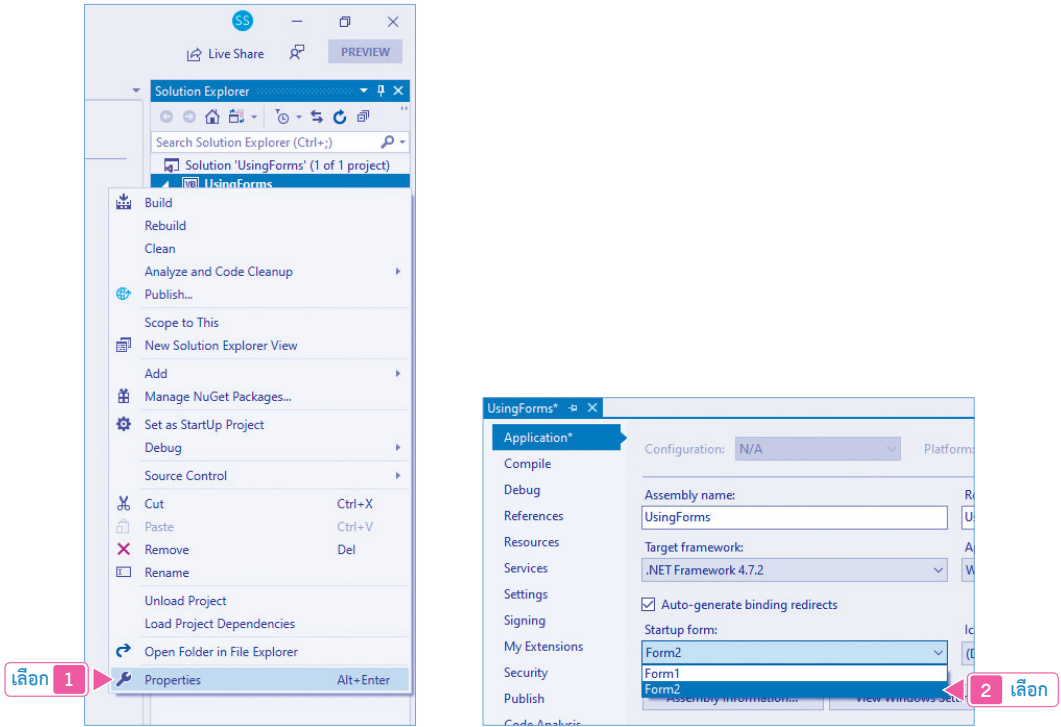
การเพิ่มฟอร์มใหม่เข้ามาในโปรเจกต์

ในการพัฒนาแอปขึ้นมาใช้งาน ย่อมที่จะต้องประกอบไปด้วยส่วนแสดงผลตั้งแต่ 2 หน้าจอขึ้นไป ส่วนวิธีการเพิ่มหน้าจอใหม่เข้ามาให้คุณคลิกเมนู Project > Add Windows Form... ในกรณีนี้เราต้องการเพิ่มหน้าจอใหม่ประเภท Windows Forms Application โดยตั้งชื่อว่า Form2 ดังรูปที่ 2-2



รูปที่ 2-2 แสดงการเพิ่มฟอร์มที่ชื่อว่า Form2 เข้ามาในโปรเจกต์

จากรูปที่ 2-2 ณ จุดนี้ โปรเจกต์ของเรามี 2 ฟอร์ม โดยปกติแล้วจะกำหนดให้ Form1 รันเป็นลำดับแรกเสมอ คุณสามารถแก้ไขได้โดยการคลิกขวาที่ชื่อโปรเจกต์ เลือก Properties กำหนดที่ช่อง Startup form: ดังรูปที่ 2-3



รูปที่ 2-9 แสดงการกำหนดให้ Form2 ปรากฏขึ้นมาเป็นลำดับแรก

วิธีการสั่งให้เปิดฟอร์มใหม่โดยการเขียนโค้ด

สำหรับวิธีการสั่งให้เปิดฟอร์มใหม่ที่เราเพิ่มเข้ามามีขั้นตอนดังนี้

1. ใน Form1 ให้คุณดับเบิลคลิกที่ปุ่มกด Button1 เพื่อสร้างเหตุการณ์ Click() จากนั้นเขียนโค้ดดังต่อไปนี้

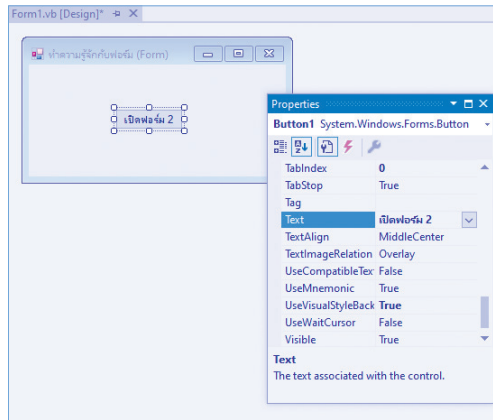
Form1.vb

```
Option Explicit On
Option Strict On

Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim frm As New Form2()
        frm.Show()
    End Sub
End Class
```

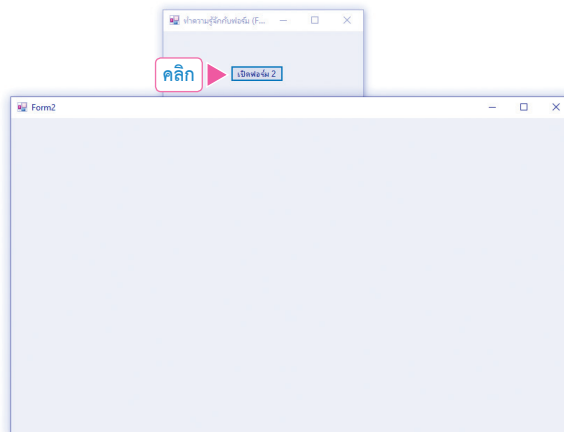
การทำงานของโค้ดข้างต้นก็คือ ในเหตุการณ์ Click() ของปุ่มกด Button ที่ชื่อว่า Button1 ให้ทำ

- สร้างตัวแปรออบเจ็กต์ Form2 ที่ชื่อว่า Dim frm As New Form2() ขึ้นมาก่อน
- สั่งให้ Form2 ปรากฏขึ้นมา ผ่านทางตัวแปร frm ร่วมกับ**เมธอด Show()** (frm.Show())



รูปที่ 2-4 แสดงโค้ดที่ใช้สำหรับเปิด Form2

2. ทดสอบรันโปรแกรม พบว่าเราสามารถเปิด Form2 ได้แล้ว ดังรูปที่ 2-5



รูปที่ 2-5 Form2 ที่ถูกเปิดจากปุ่มกด Button1 ใน Form1

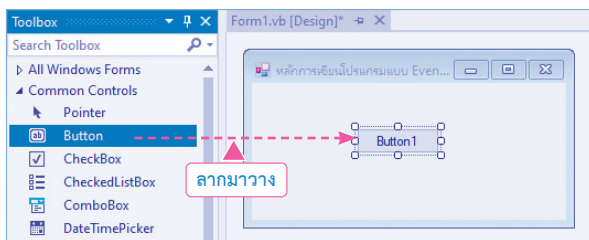


หลักการเขียนโปรแกรมแบบ Event Driven Programming

หลักการทำงานของภาษา VB เรียกว่า **การเขียนโปรแกรมตอบสนองเหตุการณ์ที่เกิดขึ้น (Event Driven Programming)** กล่าวคือ คุณเป็นผู้กำหนดเองว่า เมื่อเกิดเหตุการณ์ Click() ให้ทำอะไร, เมื่อเกิดเหตุการณ์ Load() ให้ทำอะไร เป็นต้น

หมายความว่า แท้ที่จริงแล้วการกระทำต่างๆ ที่ผู้ใช้งานกระทำต่อโปรแกรมของคุณ คุณเป็นผู้กำหนดขึ้นมาเองทั้งสิ้นว่า “ให้ทำอะไร” ประเด็นที่ตามมาก็คือ วิธีการสร้างเหตุการณ์ให้กับคอนโทรลแต่ละตัวอย่างไร

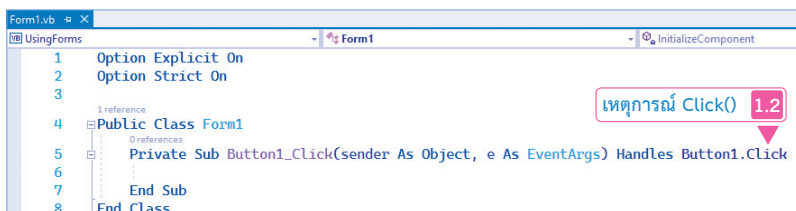
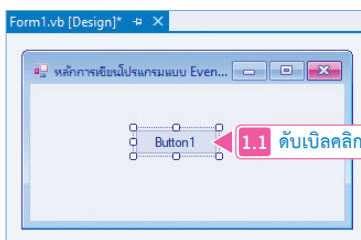
สมมติว่า ผู้เขียนต้องการสร้างเหตุการณ์ Click() ให้กับปุ่มกด Button ให้คุณลากปุ่มกด Button มาวางใน Form1 ดังรูปที่ 2-6



รูปที่ 2-6 แสดงการใช้งานคอนโทรล Button ใน Form1


การสร้างเหตุการณ์ให้กับปุ่มกด Button1 นี้ มี 2 วิธี คือ

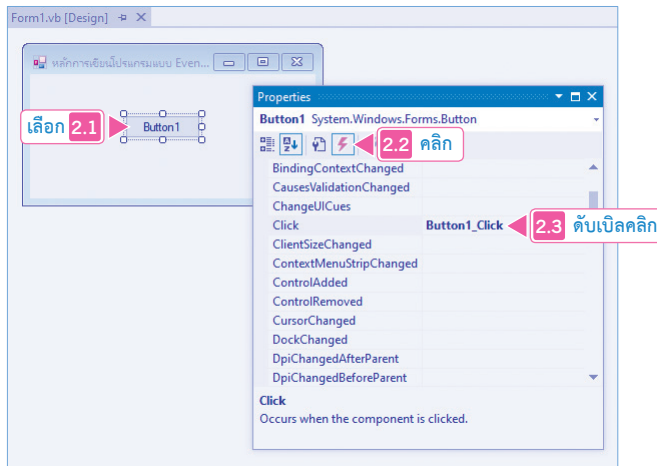
1. ดับเบิลคลิกที่คอนโทรล Button1 เพื่อสั่งให้ VS 2019 สร้างเหตุการณ์ประจำตัวของคอนโทรล Button ขึ้นมา คอนโทรล Button ถูกสร้างขึ้นมาเพื่อทำหน้าที่รองรับการคลิกจากผู้ใช้งาน ดังนั้น เหตุการณ์ประจำตัวของคอนโทรล Button ก็คือ เหตุการณ์ Click()



รูปที่ 2-7 แสดงการสร้างเหตุการณ์ Click() ด้วยวิธีดับเบิลคลิก

แต่ละคอนโทรลมีเหตุการณ์ประจำตัวแตกต่างกัน ขึ้นอยู่กับว่าคอนโทรลตัวนั้นๆ ถูกสร้างขึ้น มาเพื่อ “ทำหน้าที่อะไร”

2. การสร้างเหตุการณ์อีกวิธีหนึ่ง คือ ให้คุณโฟกัสที่คอนโทรล Button1 ก่อน จากนั้นคลิกที่ ปุ่ม  ในหน้าต่าง Properties เพื่อเลือกสร้างเหตุการณ์ที่เราต้องการ จะเห็นได้ว่าปุ่มกด Button รองรับหลายเหตุการณ์ ให้คุณดับเบิลคลิกที่ Click เพื่อสร้างเหตุการณ์ Click() ขึ้นมาสำหรับปุ่มกด Button1 ปุ่มนี้



รูปที่ 2-8 แสดงเหตุการณ์ Click() ที่ได้จากรีเสลอสสร้างเหตุการณ์

ผู้เขียนลองตั้งใจง่ายๆ ว่า ต้องการแสดงข้อความ “Hello VB” เมื่อผู้ใช้งานคลิกที่ปุ่ม Button1 ดังโค้ดต่อไปนี้

```
Form1.vb
Option Explicit On
Option Strict On

Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        MessageBox.Show("Hello VB", "คำทักทาย")
    End Sub
End Class
```

จากโค้ดข้างต้น เป็นการสั่งให้แสดงข้อความ “Hello VB” โดยอาศัยคลาส **MessageBox** เข้ามาทำหน้าที่สร้างไดอะล็อกแสดงข้อความ ดังรูปที่ 2-9



```

1 Option Explicit On
2 Option Strict On
3
4 Public Class Form1
5     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
6         MessageBox.Show("Hello VB", "คำทักทาย")
7     End Sub
8 End Class
9

```

รูปที่ 2-9 แสดงข้อความที่ได้

3. ในกรณีที่ดับเบิลคลิกบริเวณพื้นที่ว่างๆ ของ Form1 ส่งผลให้ VB สร้างเหตุการณ์ Load() ขึ้นมาให้ เหตุการณ์ประจำตัวของ Form ก็คือ เหตุการณ์ Load() เพื่อให้คุณกำหนดว่าเมื่อ Form1 ถูกโหลดขึ้นมาแล้วให้ทำอะไร

```

1 Option Explicit On
2 Option Strict On
3
4 Public Class Form1
5     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
6         MessageBox.Show("Hello VB", "คำทักทาย")
7     End Sub
8
9     Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
10
11 End Sub
12 End Class
13

```

รูปที่ 2-10 แสดงที่จัดเก็บโค้ดภาษา VB

พื้นฐานการใช้ฟอร์มแบบ MDI และสร้างระบบเมนู (Menu)

ฟอร์ม (Form) ทำหน้าที่เป็นตัวบรรจุ (Container) คอนโทรลต่างๆ ที่คุณเรียกใช้งานอยู่ โดยที่ในแอปพลิเคชันชนิด Windows Application 1 ตัว จะต้องประกอบไปด้วยฟอร์มอย่างน้อยที่สุด 1 ฟอร์ม เพื่อทำหน้าที่เป็นส่วนแสดงผลให้กับผู้ใช้ (เรียกว่า **User Interface-UI**) รูปร่างหน้าต่างโปรแกรมของคุณจะเป็นอย่างไร ก็จะขึ้นอยู่กับกรอบแบบฟอร์มของคุณนั่นเอง

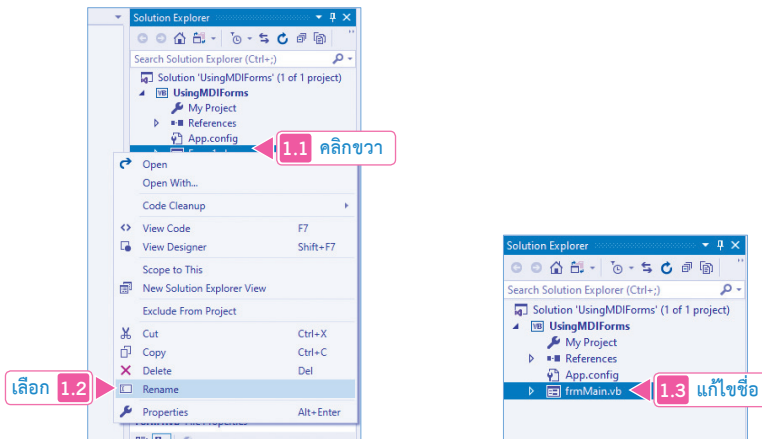
เราสามารถแบ่งลักษณะของฟอร์มได้ 2 ลักษณะ คือ

- การใช้งานฟอร์มแบบ SDI (Single Document Interfaces) หมายถึง แอปพลิเคชันที่สามารถเปิดเอกสารได้เพียงครั้งละ 1 เอกสาร หรือแสดงผลส่วนติดต่อกับผู้ใช้ได้ครั้งละ 1 หน้าจอ

- **การใช้งานฟอร์มแบบ MDI (Multiple Document Interfaces)** หมายถึง การใช้งานฟอร์มที่สามารถแสดงส่วนแสดงผล หรือส่วนติดต่อกับผู้ใช้ได้มากกว่า 1 หน้าจอในเวลาเดียวกัน โดยที่จะมี 1 ฟอร์มทำหน้าที่เป็นฟอร์มแม่ เรียกว่า **Parent Form** ทำหน้าที่บรรจุฟอร์มลูก ที่เรียกว่า **Child Form**

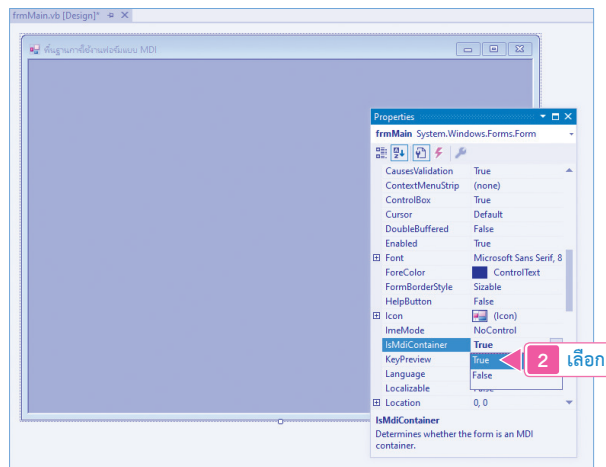
ตัวอย่างที่ 2-1 พื้นฐานการใช้งานฟอร์มแบบ MDI มีขั้นตอนดังนี้

1. สร้างโปรเจกต์ที่ชื่อว่า UsingMDIForms โดยเปลี่ยนชื่อฟอร์ม Form1 เป็น frmMain ก่อน ดังรูปที่ 2-11



รูปที่ 2-11 แสดงการเปลี่ยนชื่อฟอร์มเป็น frmMain

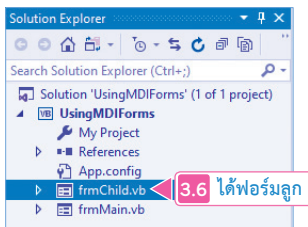
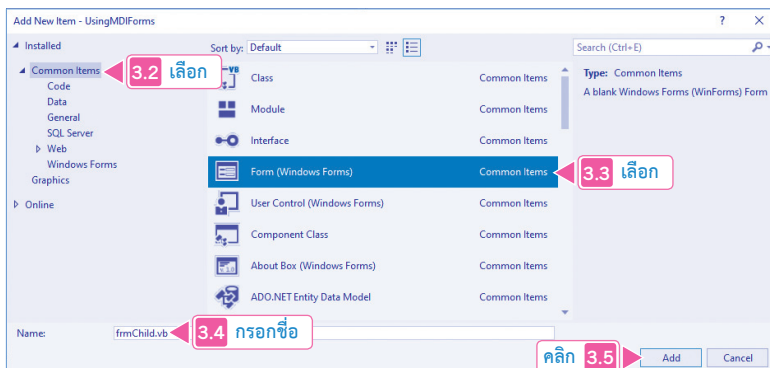
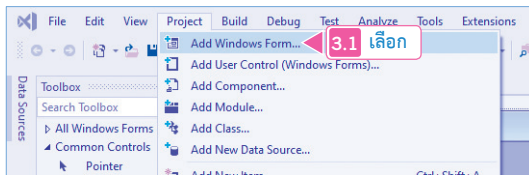
2. กำหนดให้ฟอร์มนี้เป็นฟอร์มหลัก MDI ทำหน้าที่บรรจุฟอร์มลูกต่างๆ โดยการกำหนดที่คุณสมบัติ **IsMdiContainer** = True ดังรูปที่ 2-12



รูปที่ 2-12 แสดงการกำหนดให้ฟอร์มนี้ทำหน้าที่เป็นฟอร์มหลัก MDI



3. ให้คุณเลือกเมนู Project > Add Windows Form... เพื่อเพิ่มฟอร์มใหม่เข้ามาในโปรเจกต์ และตั้งชื่อว่า frmChild ทำหน้าที่เป็นฟอร์มลูก ดังรูปที่ 2-13



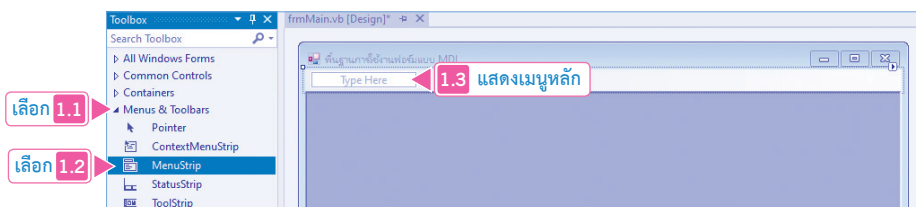
รูปที่ 2-13 แสดงฟอร์มลูกที่ถูกเพิ่มเข้ามาในโปรเจกต์

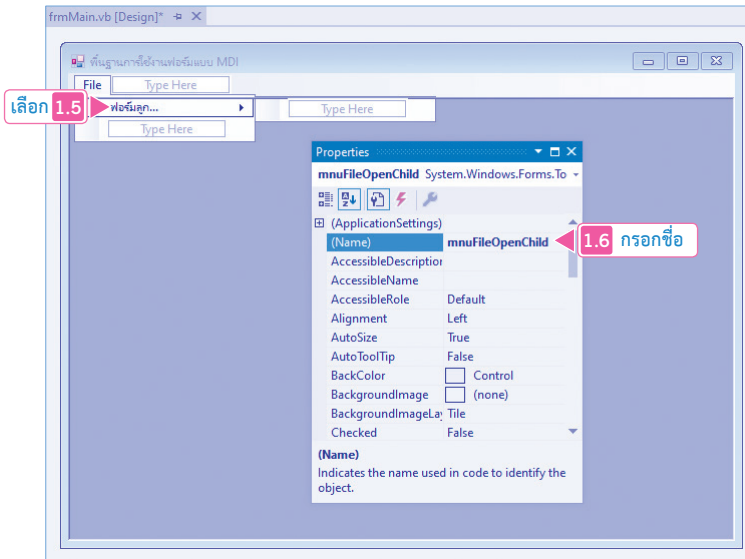
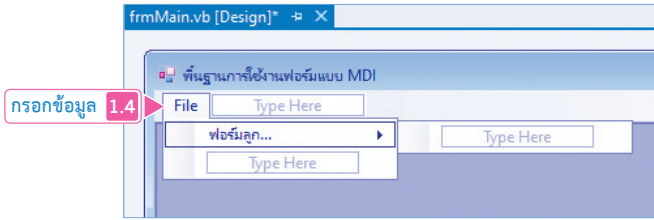
การสร้างเมนูให้กับฟอร์มหลัก

หัวข้อนี้จะเป็นการสร้างเมนูเพื่อทำหน้าที่เรียกฟอร์มลูก โดยมีขั้นตอนดังนี้

1. ที่ฟอร์ม frmMain ต้องการสร้างเมนูในโปรแกรมของเราประกอบด้วย

- **เมนู File** เป็นเมนูหลัก
- **เมนู ฟอร์มลูก...** ทำหน้าที่เปิดฟอร์มลูก





รูปที่ 2-14 แสดงการสร้างระบบเมนูในแบบฟอร์ม frmMain

2. ที่ฟอร์ม frmMain ให้คุณเขียนโค้ดดังต่อไปนี้

โค้ด VB 2019 ที่ 2-1 พื้นฐานการใช้ฟอร์มแบบ MDI (frmMain.vb)

Option Explicit On
Option Strict On

Public Class frmMain

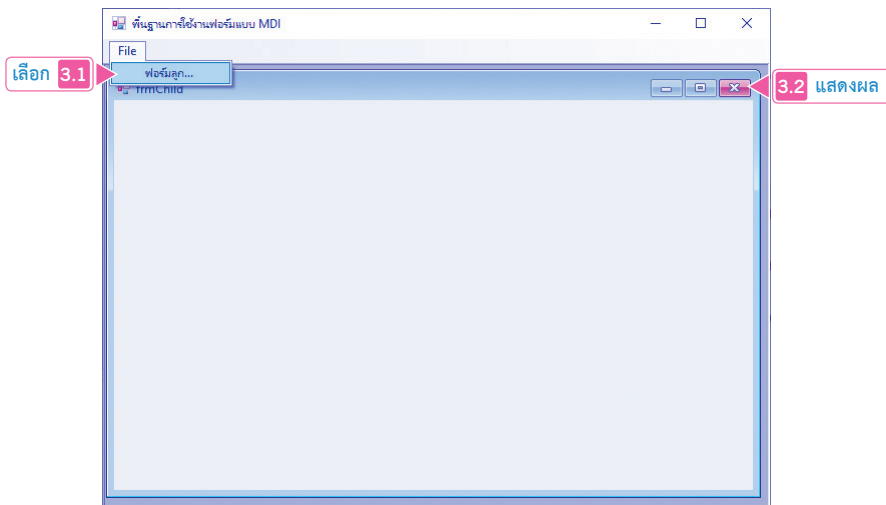
Private Sub mnuFileOpenChild_Click(sender As Object, e As EventArgs) Handles mnuFileOpenChild.Click

Dim f = New frmChild()
f.MdiParent = Me
f.Show()

End Sub
End Class



3. ท้ายที่สุด ให้คุณลองรันโปรแกรม พบว่าเมนูฟอร์มลูก... ทำหน้าที่เปิดฟอร์มลูก frmChild ขึ้นมา อยู่ในฟอร์ม frmMain ดังรูปที่ 2-15



รูปที่ 2-15 ผลการรันตัวอย่างที่ 2-1

อธิบายการทำงานของโค้ด

การเปิดฟอร์มลูกที่ชื่อว่า frmChild อยู่ในเหตุการณ์ mnuFileOpenChild_Click() โดยการสร้างตัวแปรออบเจกต์ฟอร์ม frmChild ที่ชื่อว่า f โดยที่

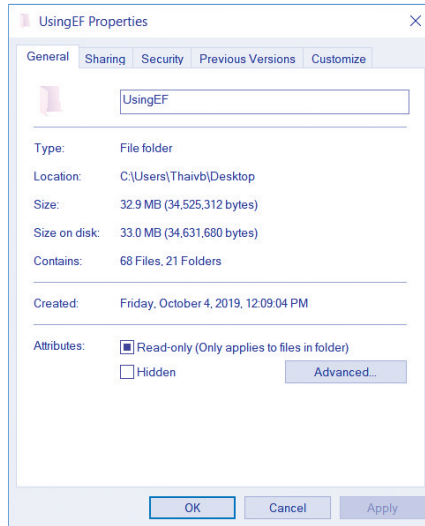
- กำหนดชื่อฟอร์มแม่ได้ที่คุณสมบัติ **MdiParent** การเรียกชื่อแทนฟอร์มแม่ frmMain โดยใช้คำสั่ง **Me**
- สั่งให้แสดงฟอร์มลูกโดยอาศัยเมธอด **Show()**

frmMain.vb

```
Private Sub mnuFileOpenChild_Click(sender As Object, e As EventArgs) Handles mnuFile
OpenChild.Click
    Dim f = New frmChild()
    f.MdiParent = Me
    f.Show()
End Sub
```

การลดขนาดโปรเจกต์ของ VB 2019

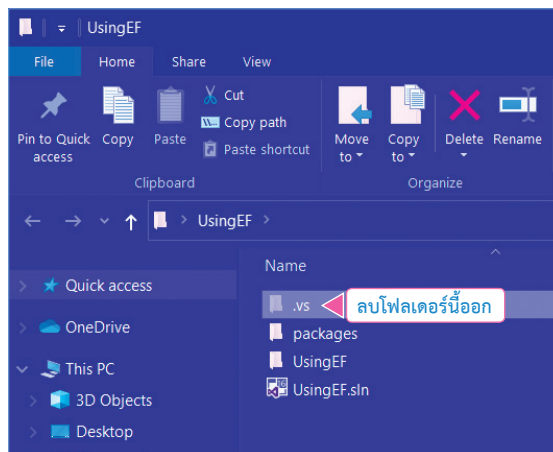
ไฟล์โปรเจกต์ที่ได้มาจากโปรแกรม Visual Studio มีขนาดค่อนข้างใหญ่มาก โดยเฉพาะอย่างยิ่งในกรณีที่มีการใช้งาน Package ต่างๆ เพิ่มเติม ดังรูปที่ 2-16



รูปที่ 2-16 แสดงขนาดโฟลเดอร์ที่จัดเก็บโปรเจกต์ VB 2019 เพียงโปรเจกต์เดียว

ในกรณีที่คุณมีโปรเจกต์จำนวนมาก ย่อมที่จะต้องใช้เวลาในการจัดเก็บมากเป็นเงาตามตัว คุณสามารถลดขนาดโปรเจกต์ลงได้ด้วย โดยมีขั้นตอนดังนี้

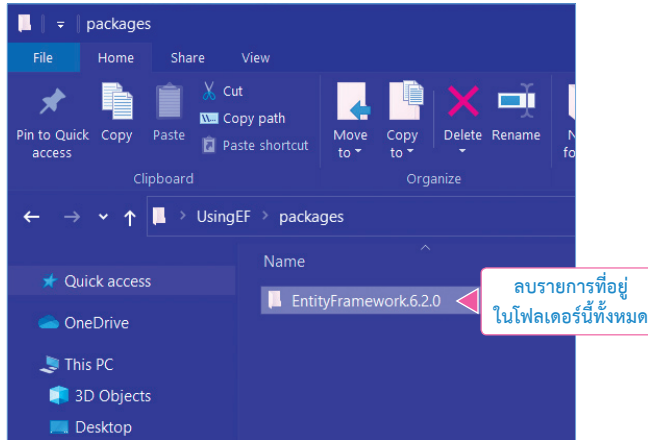
1. ให้คุณเข้าไปในโฟลเดอร์โปรเจกต์ของคุณก่อน จากนั้นลบโฟลเดอร์ `.vs` ออกได้เลย ดังรูปที่ 2-17



รูปที่ 2-17 แสดงการลบโฟลเดอร์ `.vs`

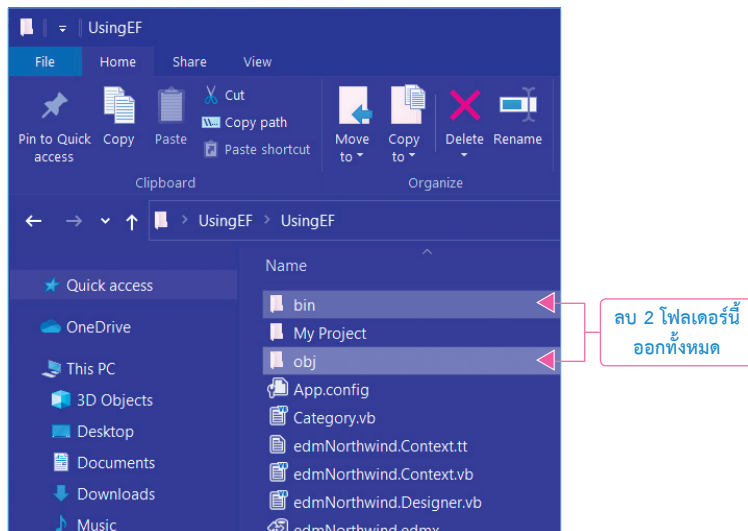


2. ในกรณีที่โปรเจกต์ของคุณมีการใช้งาน Package อื่นๆ เพิ่มเติม ก็จะมีโฟลเดอร์ที่ชื่อว่า Package เพิ่มเติมขึ้นมาให้คุณเข้าไปในโฟลเดอร์นี้ จากนั้นลบรายการไฟล์ต่างๆ ออกเช่นกัน ดังรูปที่ 2-18



รูปที่ 2-18 แสดงการลบรายการ Package เพิ่มเติมต่างๆ

3. ให้คุณเข้าไปในโฟลเดอร์โปรเจกต์อีก 1 ระดับ จากนั้นลบโฟลเดอร์ bin และ obj ออก ดังรูปที่ 2-19



รูปที่ 2-19 แสดงการลบโฟลเดอร์ bin และโฟลเดอร์ obj