



🎽 ความหมายของโปรแกรมเชิงวัตถุ

การทำโปรแกรมเชิงวัตถุ หมายถึง วิธีการเขียน โปรแกรมของนักเขียนโปรแกรมรุ่นใหม่ ที่จัดแบ่งการ เขียนคำสั่งกันออกเป็นชุด ๆ แต่ละชุดเรียกว่า "วัตถุ" (object) แล้วจึงนำเอาชุดคำสั่งแต่ละชุดนั้นมารวมกัน เป็นโปรแกรมชุดใหญ่อีกทีหนึ่ง ในบางครั้งยังอาจนำ "วัตถุ" ของโปรแกรมหนึ่งไปรวมกับ "วัตถุ" ของอีก โปรแกรมหนึ่ง แล้วเรียกออกมาใช้ได้เลย 2 🄰 บทบาทของโปรแกรมเชิงวัตถุ

วัตถุ



โปรแกรมเชิงวัตถุ OOP เป็นวิธีการเขียนโปรแกรม โดยอาศัย แนวคิดของวัตถุชิ้นหนึ่ง มีความสามารถในการปกป้องข้อมูล และการสืบ ทอดคุณสมบัติ ซึ่งทำให้แนวโน้มของOOP ได้รับการยอมรับและได้พัฒนา มาใช้ในระบบต่าง ๆ มากมาย เช่น ระบบปฏิบัติการวินโดวส์ เป็นต้น การเขียนโปรแกรมเชิงวัตถุ OOP : Object Oriented Programming คือหนึ่งในรูปแบบการเขียนโปรแกรมคอมพิวเตอร์ ที่ให้ความสำคัญกับ **แนวคิดดั้งเดิมของการเขียนโปรแกรม** ก็คือ การแก้ปัญหาโดยใช้คอม พิวเตอร์เป็นเครื่องมือ แนวคิดนี้คล้ายกับการใช้เครื่องคิดเลขในการแก้ปัญหา ทางคณิตศาสตร์

ความเป็นมาและแนวคิดการเขียนโปรแกรมเชิงวัตถุ

**แนวคิดแบบใหม่ที่ใช้ในการการเขียนโปรแกรม** ก็คือ การเน้นถึงปัญหา และองค์ประกอบของปัญหาเพื่อแก้ปัญหา การเน้นที่ปัญหาและองค์ประกอบ ของการแก้ปัญหา จะคล้ายกับการแก้ไขปัญหาและชีวิตความเป็นอยู่ของมนุษย์ ที่จะต้องมี คน สัตว์ สิ่งของ เพื่อแก้ปัญหา

อาลัน เคย์ (Alan Kay) เป็นผู้บุกเบิกแนวความคิดในการเขียนโปรแกรมเชิง วัตถุ ซึ่งเป็นต้นแบบของการเขียนโปรแกรมเชิงวัตถุได้เสนอกฎ 5 ข้อ ที่เป็น แนวทางของภาษาคอมพิวเตอร์เชิงวัตถุไว้ดังนี้

- 1. ทุก ๆ สิ่งเป็นวัตถุ
- 2. โปรแกรม ก็คือ กลุ่มของวัตถุที่ส่งข่าวสารบอกกันและกันให้ทำงาน
   3. ในวัตถุแต่ละวัตถุจะต้องมีหน่วยความจำและประกอบไปด้วยวัตถุอื่นๆ
   4. วัตถุทุกชนิดจะต้องจัดอยู่ในประเภทใดประเภทหนึ่ง
- 5. วัตถุที่จัดอยู่ในประเภทเดียวกันย่อมได้รับข่าวสารเหมือนกัน

#### แนวคิดแบบ OOP

ถ้าเราไม่มองในแง่มุมของการเขียนโปรแกรมเพียงอย่างเดียว ให้มองไปในภาพรวม มองไปในสิ่งรอบ ๆ ตัวเราสามารถบอกได้ว่าแนวคิดของ OOP ก็คือ "ธรรมชาติของวัตถุ" หมายความว่า OOP จะมองสิ่งแต่ละสิ่งถือเป็น "วัตถุชิ้นหนึ่ง"

เนื่องจากหลักการเขียนโปรแกรมเชิงวัตถุเป็นแนวคิดแบบใหม่ ดังนั้น การทำงานหลาย ๆ ส่วนของการเขียนโปรแกรมแบบนี้ ซึ่งยังไม่เป็นที่คุ้นเคยมากนัก จึงควรมีความรู้เบื้องต้น เกี่ยวกับ OOP ดังนี้ การเชื่อมต่อ การซ่อนรายละเอียด การนำวัตถุมาใช้ใหม่ การพ้องรูป การเขียนโปรแกรมและออกแบบระบบงาน เปรียบเทียบแนวคิดระหว่างการเขียนโปรแกรมเชิงกระบวนการและเชิงวัตถุ แนวทางการออกแบบและแก้ปัญหา ดีไซน์แพตเทิร์น–แบบแผนและแนวทางการออกแบบ การเขียนโปรแกรมเชิงวัตถุและฐานข้อมูล โปรแกรมเชิงวัตถุและการเทียบเคียงกับโลกของความเป็นจริง



🔪 ประโยชน์ของโปรแกรมเชิงวัตถุ

1.5

 ความสามารถในการเรียกใช้ได้หลายครั้ง ออบเจ็กต์ได้ถูกออกแบบตามหลักการที่ว่า สามารถเรียกใช้งานได้หลาย ๆ ครั้ง ในหลักการนี้ทำให้แอปพลิเคชัน ของ OOP ตัวแรกอาจจะ ทำได้ยาก แต่ว่าโปรแกรมแอปพลิเคชันที่เขียนภายหลังจะสร้างง่ายเพราะสามารถเรียกใช้ ออบเจ็กต์ที่ถูกสร้างไว้ตั้งแต่โครงงานแรกได้

2. ความเชื่อถือได้ โปรแกรมแอปพลิเคชันของ OOP จะมีความเชื่อถือได้สูง เพราะจะ รวมเอาส่วนย่อยที่ทดสอบจนได้มาตรฐานแล้วมารวมเข้าไว้ด้วยกัน รหัส (Code) ที่เขียน ขึ้นมาใหม่ในแต่ละแอปพลิเคชันจะมีไม่มากนัก

 ความต่อเนื่องกัน การพัฒนาซอฟต์แวร์แบบ OOP ใน C++ จะเปลี่ยนไปตามฝีมือ และจำนวนนักเขียนโปรแกรมภาษา C นักเขียนโปรแกรมภาษาซีสามารถรู้หลักการของ OOP ได้ภายในเวลาไม่นาน และสามารถเข้าใจเนื้อหาได้ไม่ยาก อีกทั้งสามารถแปลงโปรแกรมแอป พลิเคชันของ C เป็น C++ ได้อย่างรวดเร็ว





#### 2.1 > ภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุเบื้องต้น

- Visual Basic
- ABAP
- C#
- Javascript
- Delphi++
- JADE
- Nice
- Objective-C
- OCaml
- Perl
- Python
- Ruby
- Sleep
- Ada 95

- Java
- Fortran 2003
- C++
- Common Lisp Object System
- Eiffel
- Modula–3
- Oberon
- Objective Modula-2
- Object Pascal
- PHP
- REALbasic
- Simula
- Specman

## ประวัติของภาษาในการเขียนโปรแกรมเชิงวัตถุเบื้องต้น

#### 2.2.1 ภาษา Visual Basic

ภาษา Visual Basic พัฒนาโดย Prof. Kemeny และ Kurtz ที่เมือง Dartmouth เกิดขึ้นในปีค.ศ. 1960 โดยมีจุดประสงค์สำหรับใช้สอนในห้องคอมพิวเตอร์ เมื่อมีการ พัฒนาเครื่องไมโครคอมพิวเตอร์ขึ้นในยุคแรก ๆ จะมีหน่วยความจำไม่เพียงพอที่จะ ทำงานกับโปรแกรมภาษาอื่น เช่น Fortran และ Cobol

#### 2.2.2 <mark>ภาษา</mark> JAVA

จาวา เป็นภาษาใหม่ที่ได้รับความนิยมมาก ภาษาจาวา ได้รับการพัฒนาขึ้นโดย บริษัท ซันไมโคร ซิสเตมส์ ในปี 1991 โดยมีเป้าหมายที่จะสร้างผลิตภัณฑ์ อิเล็กทรอนิกส์ สำหรับผู้บริโภคที่ใช้ง่าย มีค่าใช้จ่ายต่ำไม่มีข้อผิดพลาด และสามารถใช้กับเครื่องใด ๆ ก็ ได้ ซึ่งสิ่งเหล่านี้ก็ได้กลายเป็นข้อดีของจาวา ที่เหนือกว่าภาษาอื่น ๆ โดยเฉพาะอย่างยิ่ง ใน การที่โปรแกรมซึ่งเขียนขึ้นด้วยจาวาสามารถนำไปใช้กับเครื่องต่าง ๆ

#### 2.2.3 ภาษาอาบัป (ABAP)

ภาษาอาบัป เป็นภาษาโปรแกรมระดับสูงที่พัฒนาโดยบริษัท SAP จากประเทศ เยอรมนี โดยเป็นภาษาที่ใช้ในซอฟต์แวร์ประเภทเว็บแอปพลิเคชันเซิร์ฟเวอร์ของบริษัท SAP เอง รูปแบบของภาษา ABAP ใกล้เคียงกับภาษาโคบอล

## 2.2.4 ภาษาฟอร์แทรน

ภาษาฟอร์แทรน หรือเป็นภาษาที่เก่าแก่ที่สุดของการคอมพิวเตอร์ ถูกพัฒนาขึ้นมา ในยุคคริสต์ทศวรรษ 1950 นิยมนำไปใช้ในการคำนวณทางคณิตศาสตร์และวิทยาศาสตร์ จนถึงปัจจุบันนี้ ภาษาฟอร์แทรนก็ยังถูกใช้ในทางวิทยาศาสตร์

### 2.2.5 ภาษาซีชาร์ป (C#)

ภาษาซีชาร์ป เป็นภาษาโปรแกรมแบบหลายโมเดล ที่ใช้ระบบชนิดข้อมูลแบบ รัดกุม และสนับสนุนการเขียนโปรแกรมเชิงคำสั่ง การเขียนโปรแกรมเชิงประกาศการ เขียนโปรแกรมเชิงฟังก์ชัน การเขียนโปรแกรมเชิงกระบวนการ การเขียนโปรแกรมเชิง วัตถุ (แบบคลาส)และการเขียนโปรแกรมเชิงส่วนประกอบ

#### 2.2.6 Visual C#

Visual C# (วิชวล-ซี-ชาร์ป) เป็นโปรแกรมหรือเครื่องมือที่ช่วยในการเขียนโปรแกรม คอมพิวเตอร์ด้วยภาษา C# ซึ่งโปรแกรมนี้ได้พัฒนาขึ้นครั้งแรกในชื่อของ Visual C# .NET (วิชวล-ซี-ชาร์ป-ดอตเน็ต) ต่อมาก็ได้พัฒนาโปรแกรมให้มีความสามารถสูงขึ้นและรองรับการ เขียนโปรแกรมในรูปแบบต่าง ๆมากมาย

### 2.2.7 ภาษาซีพลัสพลัส (C++)

ภาษาซีพลัสพลัส เป็นภาษาโปรแกรมคอมพิวเตอร์อเนกประสงค์ มีโครงสร้างภาษาที่มีการ จัดชนิดข้อมูลแบบสแตติก และสนับสนุนรูปแบบการเขียนโปรแกรมที่หลากหลาย ภาษาซีพลัสพ ลัสเป็นภาษาโปรแกรมเชิงพาณิชย์ที่นิยมมากภาษาหนึ่ง นับตั้งแต่ช่วงทศวรรษ 1990 เบียเนอ สเดราสดร็อบ จากเบลล์แล็บส์ เป็นผู้พัฒนาภาษาซีพลัสพลัส (เดิมใช้ชื่อ "C with classes") ในปี ค.ศ. 1983 เพื่อพัฒนาภาษาซีดั้งเดิมสิ่งที่พัฒนาเพิ่มเติมนั้น เริ่มจากการเพิ่มเติมการสร้างคลาส จากนั้นก็เพิ่มคุณสมบัติต่าง ๆ ตามมา ซึ่งได้แก่เวอร์ชวลฟังก์ชัน การโอเวอร์โหลดโอเปอร์เรเตอร์ การสืบทอดหลายสาย เทมเพลต และการจัดการเอกเซพชันมาตรฐานของภาษาซีพลัสพลัสได้รับ การรับรองในปี ค.ศ. 1998 เป็นมาตรฐาน ISO/IEC 14882:1998 เวอร์ชันล่าสุด คือ C++ 2019





# ความรู้พื้นฐานเกี่ยวกับภาษา Visual C#

#### 3.1.1 ชนิดข้อมูล (Data Type)

3.1

ในภาษาซีชาร์ป ได้มีการกำหนดชนิดของข้อมูล ไว้หลากหลายชนิด เพื่อสามารถรองรับ ข้อมูลหลาย ๆ ประเภท ดังตาราง

ชนิดข้อมูล	คำอธิบาย	ขอบเขตข้อมูล
sbyte	จำนวนเต็มทั้งบวกและลบ ขนาด 8 บิต	–128 ถึง 127
short	จำนวนเต็มทั้งบวกและลบ ขนาด 16 บิต	–32,768 ถึง 32,768
int	จำนวนเต็มทั้งบวกและลบ ขนาด 32 บิต	–2,147,483,648 ถึง 2,147,483,647
long	จำนวนเต็มทั้งบวกและลบ ขนาด 64 บิต	-2 <sup>63</sup> ถึง 2 <sup>63</sup> -1
byte	จำนวนเต็มบวก ขนาด 8 บิต	0–255
ushort	จำนวนเต็มบวก ขนาด 16 บิต	0–65,535
uint	จำนวนเต็มบวก ขนาด 32 บิต	0–4,294,967,295
ulong	จำนวนเต็มบวก ขนาด 64 บิต	0-2 <sup>64</sup> -1
float	จำนาวนนุกสนิยน อนากล 22 มิด	ค่าลบ –3.4 x 10 <sup>38</sup> ถึง –1.4 x 10 <sup>45</sup>
noat	ง เน่านทศนยม ชน เด ว2 บด	ค่าบวก –1.4 × 10 <sup>45</sup> ถึง 3.4 × 10 <sup>38</sup>
double	จำนานแต็นนาก อนาก 64 นิต	ค่าลบ –1.8 × 10 <sup>308</sup> ถึง –4.9 × 10 <sup>-324</sup>
double		ค่าบวก −1.4 × 10 <sup>45</sup> −3.4 × 10 <sup>38</sup>
bool	ข้อมูลชนิดตรรกะ	มี 2 ค่า คือ ค่าจริง (true) และค่าเท็จ (false)
char	ข้อมูลชนิดตัวอักษรตัวเดียว	เช่น 'A', '1'
string	ข้อมูลที่ประกอบด้วยตัวอักษรหลายตัว	เช่น "Hello"

# 3.1.2 ตัวแปร (Variables)

ตัวแปร เป็นการอ้างถึงข้อมูล โดยค่าของตัวแปร สามารถเปลี่ยนแปลงได้ ตลอดเวลาที่โปรแกรมกำลังทำงานอยู่ ในภาษาซีชาร์ป ตัวแปรทุกตัวต้องถูกประกาศ ก่อนที่จะถูกนำมาใช้งาน โดยมีการระบุชนิดข้อมูลที่จะใช้กับตัวแปรนั้น ๆ ไว้

 ภาษาซีชาร์ป มีกฎเกณฑ์การตั้งชื่อตัวแปร ดังนี้
 ⇒ ตัวอักษรตัวแรกของชื่อต้องเป็นตัวอักษรภาษาอังกฤษ หรือตัวขีด เส้นใต้ (\_)
 ⇒ ชื่อตัวระบุต้องประกอบด้วยตัวอักษรภาษาอังกฤษ (A–Z,a–z)
 ตัวเลข (0–9) หรือ
 เครื่องหมายขีดเส้นใต้ (\_) เท่านั้น
 ⇒ ชื่อตัวระบุต้องไม่ซ้ำกับคำสงวน (reserved word)

# คำสงวน มีดังต่อไปนี้

abstract	as	base	bool	break	byte	case
catch	char	checked	class	const	continue	Decimal
default	delegate	do	double	else	enum	Event
explicit	extern	false	finally	fixed	float	For
foreach	get	goto	if	implicit	in	int
interface	internal	is	lock	lonk	namespace	New
null	object	operator	out	override	params	Partial
private	protected	public	readonly	ref	return	Sbyte
sealed	set	Short	Sizeof	Stackalloc	Static	String
Sruct	Switch	This	Throw	True	Try	Typeof
Unit	Ulong	Unchecked	Unsafe	Ushort	Using	Value
Virtual	Void	Volatile	Where	while	yield	

\*

#### การประกาศใช้ตัวแปร

รูปแบบการประกาศตัวแปร ตัวอย่างการประกาศตัวแปรมีดังนี้ ชนิดของข้อมูล ชื่อตัวแปร;

### ตัวอย่างการประกาศค่าตัวแปรมีดังนี้

double x, y;

X และ y เป็นชื่อของตัวแปร ที่มีชนิดข้อมูลเป็นจำนวนทศนิยม

### การเขียนคำอธิบายโปรแกรม

ในการเขียนโปรแกรมควรมีการเขียนข้อความที่อธิบายการทำงานของโปรแกรมไว้ ด้วย เพื่อให้ผู้เขียนโปรแกรมสามารถกลับมาอ่านได้ ในกรณีที่เขียนโปรแกรมขนาด ใหญ่ ซึ่งการเขียนคำอธิบายทำได้ 2 วิธีดังนี้

1. ใช้เครื่องหมาย /\* เพื่อเปิด และปิดด้วยเครื่องหมาย \*/

2. ใช้เครื่องหมาย // นำหน้าข้อความที่อธิยาย ไปจนถึงสิ้นสุดบรรทัดนั้น

# 3.1.3 ค่าคงที่ (Constants)

ค่าคงที่เป็นอ้างถึงข้อมูลเช่นเดียวกับตัวแปร สิ่งที่แตกต่างจากตัวแปร คือ ค่าของ มันไม่สามารถเปลี่ยนแปลงได้อีกหลังจากการประกาศ

ในภาษาซีชาร์ป ค่าคงที่ต้องถูกประกาศโดยระบุชนิดข้อมูลและค่าตั้งตนก่อนถูก นำมาใช้งานเสมอการประกาศค่าคงที่จะคล้ายคลึงกับการประกาศตัวแปร แตกต่างกัน ตรงที่ต้องมีการะบุด้วยคีย์เวิร์ด const

<mark>รูปแบบคำสั่ง</mark> const ชนิดของข้อมูล ชื่อค่าคงที่ = ค่าที่ค่าคงที่ถูกใช้เป็นตัวแทน;

> const int a = 1; /\* a เป็นชื่อของค่าคงที่ ที่มีค่าเท่ากับ 1 มีชนิดข้อมูลเป็นเลขจำนวนเต็ม \*/

#### ตัวดำเนินการทางคณิตศาสตร์

ในภาษาซีชาร์ป มีตัวอย่างตัวดำเนินการทางคณิตศาสตร์ ดังนี้

ตัวดำเนินการทางคณิตศาสตร์	คำอธิบาย	ตัวอย่าง
+	บวก	z = x + y; //z คือ ค่า x บวก y
-	ลบ	z = x – y; //z คือ ค่า x ลบ y
*	คูณ	z = x * y; //z  คือ ค่า x คูณ y
/	หาร	z = x / y; //z คือ ค่า x หาร y
%	เศษจากการหาร	z = x % y; // z คือ เศษที่เหลือจาก x หาร y

# ตัวดำเนินการเปรียบเทียบ

ตัวดำเนินการทางตรรกศาสตร์ใช้เปรียบเทียบข้อมูลด้านตรรกศาสตร์ ซึ่งผลลัพธ์ที่ได้ จะเป็นจริง(True) หรือเท็จ (False) เท่านั้น มีดังนี้

ตัวดำเนินการทางคณิตศาสตร์	คำอธิบาย	ตัวอย่าง
I	ณิสะ (NOT)	!p เป็นจริง เมื่อค่า p เป็นเท็จ และเป็นเท็จเมื่อค่า p
1	นเสอ (INOT)	เป็นจริง
&&	และ (AND)	P && q เป็นจริงเมื่อ p และ q เป็นจริง
	หรือ (OR)	P    q เป็นจริงเมื่อ p และ q เป็นเท็จ
٨	Exclusive OR	P ^ q เป็นจริงเมื่อ p และ q มีค่าต่างกัน

### ตัวดำเนินการในการกำหนดค่า

ตัวดำเนินการในการกำหนดค่า สามารถใช้ในการกำหนดค่าให้กับตัวแปร ตัวดำเนินการ ในการกำหนดค่าดังตารางต่อไปนี้

ตัวดำเนินการใน การกำหนดค่า	คำอธิบาย	ตัวอย่าง
=	กำหนดค่า	x = y; หมายถึง นำค่า y ให้กับ x
+=	เพิ่มค่าแล้วกำหนดค่า	x += y; มีค่าเท่ากับ x = x + y;
-=	ลดค่าแล้วกำหนดค่า	x -= y; มีค่าเท่ากับ x = x – y;
*=	คูณแล้วกำหนดค่า	X *= y; มีค่าเท่ากับ x = x * y;
/=	หารแล้วกำหนดค่า	X /= y; มีค่าเท่ากับ x = x / y;
%=	หาเศษจากการหารแล้วกำหนดค่า	X %= y; มีค่าเท่ากับ x = x % y;

### ตัวดำเนินการอื่น ๆ

นอกจากตัวดำเนินการที่กล่าวมาแล้ว ภาษาซีชาร์ป ยังมีตัวดำเนินการอื่น ๆ อีก เช่น

- 🔶 + สำหรับ string คือ การนำ string 2 ค่ามาต่อกัน (concatenate)
- ➡ ++ เป็นตัวดำเนินการเพิ่มค่า โดยที่ x++ เทียบเท่ากับ x=x+1
  - . เป็นตัวดำเนินการลดค่า โดยที่ x– เทียบเท่ากับ x=x–1

#### การแปลงชนิดข้อมูล

เนื่องจากภาษาซีชาร์ปเข้มงวดเรื่องชนิดของข้อมูลมาก ซึ่งเราไม่สามารถนำข้อมูลต่างชนิดกัน มาใช้ร่วมกันได้ แต่ต้องนำมาทำการแปลงให้เป็นข้อมูลชนิดเดียวกันก่อน จึงจะสามารถนำมาใช้ งานร่วมกันได้

เมธอด	ความหมาย	ตัวอย่าง
int.Parse (ข้อความ);	แปลงข้อมูลชนิดตัวอักษร (string) ให้	i = Parse(TextBox1.Text); หมายถึง
	เป็นข้อมูลชนิดตัวเลขจำนวนเต็ม (int)	แปลงข้อมูลที่รับจาก TextBox1 ซึ่ง
		เป็นข้อมูลชนิด string ให้เป็นข้อมูล
		ชนิดจำนวนเต็ม แล้วเก็บค่าในตัวแปร i
		หรือ i = int.Parse("22"); หมายถึง
		แปลงข้อมูล "22" ซึ่งเป็นข้อมูลชนิด
		string ให้เป็นตัวเลข 22 แล้วเก็บค่า
		ในตัวแปร i
double.Parse(ข้อความ);	แปลงข้อมูลชนิดตัวอักษร (string) ให้	i=double.Parse(textBox1.Text);
	เป็นข้อมูลชนิดตัวเลขทศนิยม (double)	หมายถึงแปลงข้อมูลที่รับจาก
		textBox1 ซึ่งเป็นข้อมูลชนิด string ให้
		เป็นข้อมูลชนิดทศนิยม แล้วเก็บค่าในตัว
		แปร i หรือ
		i=double.Parse("22.45"); หมายถึง
		แปลงข้อมูล "22.45" ซึ่งเป็นข้อมูล
		ชนิด string ให้เป็นค่าตัวเลข 22.45
		แล้วเก็บค่าในตัวแปร i

#### ตารางการแปลงชนิดข้อมูล

เมธอด	ความหมาย	ตัวอย่าง
ToString();	แปลงข้อมูลประเภทใดๆ ให้เป็นข้อมูล	int a = 20;
	ชนิดตัวอักษร (string)	textBox1.Text = a.ToString();
		หมายถึง แปลงตัวเลข 20 ซึ่งเป็นข้อมูล
		ชนิดจำนวนเต็ม (int) ให้เป็นข้อมูลชนิด
		ตัวอักษร แล้วแสดงผลใน textbox1
		(textbox จะแสดงผลได้เฉพาะข้อมูล
		ชนิด string)
DateTime.Now.Tostring();	แปลงข้อมูลชนิดวันเวลา (DateTime)	<pre>Str1=DateTime.Now.Tostring();</pre>
	ให้เป็นข้อมูลชนิดตัวอักษร (string)	หมายถึง แปลงข้อมูลวันเวลาปัจจุบัน
		(DateTime.Now) ให้เป็นข้อมูลชนิด
		ตัวอักษร แล้วเก็บค่าในตัวแปร str1

# 3.1.4 การเขียนโปรแกรมแบบลำดับ

การเขียนโปรแกรมแบบโครงสร้างเป็นการเขียนโปรแกรมที่มีรูปแบบซัดเจน ไม่ซ้ำซ้อน แบ่งออกเป็น 3 รูปแบบ คือแบบลำดับ แบบเลือก และแบบวนซ้ำ ในที่นี้ จะกล่าวถึงการเขียนโปรแกรมแบบลำดับการเขียนโปรแกรมแบบลำดับมีลักษณะ การทำงานตามลำดับก่อนหลังของคำสั่งที่เขียนไว้

# ตัวอย่างสัญลักษณ์ผังงานที่ใช้ในการเขียนผังงานแบบลำดับ



# 3.1.5 **ฟังก์ชัน**

ฟังก์ชัน (Function) คือ กลุ่มของคำสั่งที่นำมาเรียงต่อกัน เพื่อการทำงานอย่าง ใดอย่างหนึ่งสามารถเรียกใช้ซ้ำ ๆ กันได้ไม่จำกัดจำนวนครั้ง ฟังก์ชันจะช่วยให้เราแบ่ง งานออกเป็นงานย่อย ๆ

เมธอด	ความหมาย	ตัวอย่าง
Math.Pow (ฐาน, เลขชี้กำลัง)	หาค่ายกกำลัง (Power)	Math.Pow(x,2) หมายถึงหาค่า x <sup>2</sup>
Math.Sqrt (ตัวเลขทศนิยม)	หาค่ารากที่สอง	Math.Sqrt(x) หมายถึงหาค่ารากที่สอง
	(Square Root)	ของ ×
Math.Abs (ตัวเลขทศนิยม)	หาค่า Absolute	Math.Abs(x) หมายถึง หาค่า  x
Math.PI	หาค่า¶	Math.Pl มีค่า 22 ÷ 7 นั่นเอง
Math.Sin (มุมที่มีค่าเป็นเรเดียน	หาค่า sine	Math.Sin(x) หมายถึง หาค่า sin(x)
Math.Cos (มุมที่มีค่าเป็นเรเดียน	หาค่า cosine	Math.Cos(x) หมายถึง หาค่า cos(x)
MathTan (มุมที่มีค่าเป็นเรเดียน)	หาค่า Tangent	Math.Tan(x) หมายถึง หาค่า Tan(x)
Math.BigMul (จำนวนเต็มตัวที่ 1,	หาผลคูณของตัวเลข 2 ตัว	Math.BigMul(x,y) หมายถึง หาค่าผลคูณ
จำนวนเต็มตัวที่ 2)		ของ x กับ y (x x y)
Math.Ceiling (ตัวเลขที่มีทศนิยม)	หาค่าจำนวนเต็มที่ถูกปัด	Math.Ceiling(10.02) จะได้ผลลัพธ์ คือ
	ขึ้นมาจากการมีทศนิยม	11 จะปัดขึ้นทั้งหมดโดยไม่สนใจทศนิยม
Math.Floor (ตัวเลขที่มีทศนิยม)	หาค่าจำนวนเต็มที่ถูกปัดลง	Math.Floor(10.99) จะได้ผลลัพธ์คือ 10
	จากการมีทศนิยม	จะปัดลงทั้งหมดโดยไม่สนใจทศนิยม
Math.Truncate (ตัวเลขที่มี	หาค่าจำนวนเต็มที่ถูกปัดลง	Math.Truncate(10.99) จะได้ผลลัพธ์คือ
ทศนิยม)	จากการมีทศนิยม	10 เช่นเดียวกับ
		Math.Floor(10.99) จะปัดทศนิยมทิ้งไป
		เหลือแต่จำนวนเต็ม
Math.Round (ตัวเลขที่มีทศนิยม)	หาจำนวนเต็ม โดยปัดขึ้น	Math.Round(10.4) ผลลัพธ์ คือ 10
	เมื่อมีค่า >0.5 แต่ถ้าเลขที่	Math.Round(10.7) ผลลัพธ์ คือ 10
	เป็นจุดทศนิยมมีค่า 0.5	Math.Round(10.5) ผลลัพธ์ คือ 10
	พอดีจะขึ้นกับตัวเลขจำนวน	Math.Round(11.3) ผลลัพธ์ คือ 11
	เต็มว่าเป็นเลขคู่หรือเลขคื่	Math.Round(11.5) ผลลัพธ์ คือ 12
	หากเป็นเลขคู่	Math.Round(11.6) ผลลัพธ์ คือ 12
	จะปัด .5 ทิ้งไป แต่ถ้าเป็น	
	เลขคี่ จะปัด 5 ขึ้น	

**พังก์ชันทางคณิตศาสตร์** ฟังก์ชันทางคณิตศาสตร์ คือ ฟังก์ชันที่ใช้ในการคำนวณค่า ทางคณิตศาสตร์อย่างใดอย่าง หนึ่งโดยมีเมธอดที่ใช้ดังนี้

เมธอด	ความหมาย	ตัวอย่าง
Math.Round (ตัวเลขที่มีทศนิยม,	หาค่าจำนวนทศนิยมที่มี	Math.Round(23.1243565,2) จะได้
จำนวนตำแหน่งทศนิยมที่ต้องการ)	ตำแหน่งทศนิยมตาม	ผลลัพธ์ 23.12
	ต้องการ	Math.Round(62.57879821,4) จะได้
		ผลลัพธ์ คือ 62.5778
		Math.Round(11.555,2) จะได้ผลลัพธ์
		คือ 11.56 (เนื่องจากตัวก่อนหน้าเป็น
		เลขคี่จึงปัดไปหาเลขคู่)
		Math.Round(11.565,2) จะได้ผลลัพธ์
		คือ 11.56 (เนื่องจากตัวก่อนหน้าเป็นเลข
		คู่จึงปัดทิ้ง)

**ฟังก์ชันเกี่ยวกับ String** ฟังก์ชันเกี่ยวกับ String คือ ฟังก์ชันที่ใช้ในการจัดกาเกี่ยวกับ ข้อความอย่างใดอย่างหนึ่ง โดย มีเมธอดที่ใช้ดังนี้

เมธอด	ความหมาย	ตัวอย่าง		
Length	นับจำนวนอักขระภายใน String	len1 = str1.Length;		
		len1 คือจำนวนตัวอักขระทั้งหมดของ str1		
ToLower	เปลี่ยนตัวอักษรให้เป็นตัวอักษรเล็กทั้งหมด	<pre>str2 = str1.ToLower();</pre>		
		str2 คือผลลัพธ์ที่เป็นอักษรตัวเล็กทั้งหมด		
ToUpper	เปลี่ยนตัวอักษรให้เป็นอักษรใหญ่ทั้งหมด	<pre>str2 = str1.ToUpper();</pre>		
		str2 คือผลลัพธ์ที่เป็นอักษรตัวใหญ่ทั้งหมด		
StartsWith	ตรวจสอบว่าเริ่มต้นด้วยข้อความที่กำหนด	result1 = str1.StartsWith(str2);		
	หรือไม่	result1 คือผลลัพธ์ที่		
		1. ให้ผลเป็น True หาก str1 ขึ้นต้นด้วย str2		
		2. ให้ผลเป็น False หาก str1 ไม่ขึ้นต้นด้วย		
		str2		

เมธอด	ความหมาย	ตัวอย่าง
เมธอด Compare	ความหมาย ตรวจสอบ String 2 ค่า ว่าเหมือนกันหรือไม่	<ul> <li>ตวอยาง</li> <li>result1 =String.Compare(str1,str2);</li> <li>result1 คือผลลัพธ์ที่</li> <li>1. ให้ผลลัพธ์เป็น 1 เมื่อ str1 มีค่ารหัส</li> <li>แอสกี้มากกว่า str2</li> <li>2. ให้ผลลัพธ์เป็น 0 เมื่อ str1 มีค่ารหัสแอสกี</li> <li>เท่ากัน str2 (เป็นข้อความที่เหมือนกันทุก</li> <li>ตัวอักษร)</li> <li>3. ให้ผลลัพธ์เป็น -1 เมื่อ str1 มีค่ารหัส</li> </ul>
Replace	แทนที่ String ด้วย String อีกตัว ใน String ที่ระบุ	แยสกานขยาวารเร2 str4 = str1.Replace(str2,3); Str4 คือผลลัพธ์ที่เปลี่ยนข้อความใน str1 ในจุดที่ มีข้อความตรงกับ str2 ให้เปลี่ยนเป็น str3 หรือ str2 = str1.Replace("in","me); str2 คือผลลัพธ์ที่เปลี่ยนข้อความใน str1 ในจุดที่ มีข้อความว่า in ให้เปลี่ยนเป็น me
Remove	ตัด String ในตำแหน่งที่ต้องการทิ้ง	<pre>str2 = str1.Remove(3); str2 คือผลลัพธ์ที่ตัดข้อความตั้งแต่ตำแหน่งที่ 4 ทิ้งไป str2 = str1.Remove(4,2); str2 คือผลลัพธ์ที่ตัดข้อความตำแหน่งที่ 5 และ 6 ออกไป</pre>
Insert	แทรก String ในตำแหน่งที่ต้องการเข้าไป	<pre>str3 = str1.Insert(3,str2); Str3 คือผลลัพธ์ที่แทรก str2 ในตำแหน่งที่ 4 ของ str1 หรือ str2 = str1.insert(3,"im"); str2 คือผลลัพธ์ที่แทรกข้อความว่า im ลงใน ตำแหน่งที่ 4 ของ str1</pre>

.

\*

4 4

-

# ฟังชันเกี่ยวกับเวลา

ฟังก์ชันเกี่ยวกับเวลา คือ ฟังชันที่ใช้ในการจัดการเกี่ยวกับเวลา โดยมีเมธอดที่ใช้ดังนี้

เมธอด	ความหมาย	ตัวอย่าง
DateTime.Now	ให้ค่าทั้งวันเดือนปี และเวลา	Date.Time dt;
	ปัจจุบัน ตามนาฬิกาของเครื่อง	Dt=DateTime.Now; หมายถึง
		ให้นำค่าวนเดือนปีและเวลา
		ปัจจุบันมาเก็บไว้ในตัวแปร dt
		เช่น 16/10/2562 16:38:10
DateTime.Now.ToShortDateString()	เป็นการหาวันเดือนปีปัจจุบัน	ถ้าเดือนปีและเวลาปัจจุบันคือ
	ตามนาฬิกาของเครื่องแบบ	16/10/2562 16:38:10
	ShortDate	จะได้ 16/10/2562
DateTime.Now.ToLongDateString();	เป็นการหาวันเดือนปีปัจจุบัน	ถ้าวันเดือนปีและเวลาปัจจุบันคือ
	ตามนาฬิกาของเครื่องแบบ	16/10/2562 16:38:10 จะได้
	LongDate	16 พฤศจิกายน 2562

เมธอด	ความหมาย	ตัวอย่าง	
DateTime.Now.ToShortTimeString()	เป็นการหาเวลาปัจจุบันตาม	ถ้าวันเดือนปีและเวลาปัจจุบันคือ	
	นาฬิกาของเครื่องแบบ	16/10/2562 16:38:10 จะได้	
	ShortTime	16:38	
DateTime.Now.ToLongTimeString()	เป็นการหาเวลาปัจจุบันตาม	ถ้าวันเดือนปีและเวลาปัจจุบันคือ	
	นาฬิกาของเครื่องแบบ	16/10/2562 16:38:10 จะได้	
	LongTime	16:38:10	
DateTime.Now.ToshortDateString()	เป็นการหาเวลาปัจจุบันตาม	ถ้าวันเดือนปีและเวลาปัจจุบันคือ	
	นาฬิกาของเครื่องแบบ	16/10/2562 16:38:10 จะได้	
	ShortDate	16/10/2562	
DateTime.Now.ToLongDateString()	เป็นการหาเวลาปัจจุบันตาม	ถ้าวันเดือนปีและเวลาปัจจุบันคือ	
	นาฬิกาของเครื่องแบบ	16/10/2562 16:38:10 จะได้	
	LongDate	16 พฤศจิกายน 2562	



โปรแกรมภาษา C# ขั้นพื้นฐานที่มีเฉพาะส่วนของโปรแกรมหลัก และไม่มี โปรแกรมย่อย จะมีส่วนประกอบดังนี้



วิธีติดตั้ง Visual Studio 2010

 1. ใส่แผ่น DVD โปรแกรมจะ มี AutoRun ดังหมายเลข 1 ให้ เลือก หรือเลือก Setup.exe ดัง หมายเลข 2



Irganize  Burn to d	isc						
Favorites	Files Currently on the D	Disc (81) -					Files Ready to Be Written to th
Desktop E	JProductDocumentation	acab10	Cab24	Cab38	cab52	cab66	@ desktop
Downloads	Remote Debugger	Cab11	Cab25	Cab39	cab53	cab67	
Secent Places	La Setup	Cab12	acab26	Cab40	cab54	cab68	
SkyDrive	Standalone Profiler	Cab13	Cab27	cab41	cab55	autorun	
	MCU WCU	Cab14	cab28	Cab42	cab56	autorun	
Libraries	ab1	Cab15	Cab29	Cab43	cab57	htmllite.dll	[2]
Documents	Cab2	Cab16	cab30	Cab44	cab58	locdata	
Music	Cab3	Cab17	Cab31	Cab45	cab59	🗑 readme	
Pictures	ab4	acab18	Cab32	cab46	cab60	setup 4	
Videos	cab5	ab19	Cab33	Cab47	Cab61	setup	
	ab6	ab20	cab34	Cab48	Cab62	Vs_setup	
Computer	Cab7	Cab21	cab35	Cab49	cab63	-	
Local Disk (C:)	ab8	Cab22	cab36	cab50	Cab64		
Local Disk (D:)	Cab9	Cab23	Cab37	Cab51	Cab65		
DVD RW Drive (E _		-	-	_	-		



เอาเครื่องหมายถูกหน้า Yes, send
 information ออก แล้วกดปุ่ม Next เพื่อทำ
 การติดตั้งดังหมายเลข 4





4. เลือก I have read and accept the License terms ดังหมายเลข 5 เลือก Next ดังหมายเลข 6



# 5. เลือกการติดตั้งแบบ Full แล้วเลือก Install ดังหมายเลข 7 และ หมายเลข 8

 6. โปรแกรมจะทำการติดตั้ง Visual Studio ใช้เวลานาน (ประมาณ 30 นาที) ติดตั้งเสร็จแล้วจะขึ้นหน้าจอนี้ให้เลือก Finish ดังหมายเลข 9 หมายเลข 10 และ หมายเลข 11
## 7. ทดสอบเปิดโปรแกรมโดยเลือก Visual Basic Development Setting



General Development Settings Project Management Settings Visual Basic Davalopment Settings Visual C+ Development Settings Visual F# Development Settings Web Development (Code Only)

Start Visual Studio

.

Optimizes the environment so you can focus on building world-class applications. This collection of settings contains customizations to the window layout, command menus and keyboard shortcuts to make common Visual Basic commands more accessible.

**Exit Visual Studio** 

licrosoft	Visual Studio
1	Microsoft Visual Studio is loading user settings. This may take a few minutes.

2 B B B

0

1

-

8. เลือก New Project เลือก Windows Forms Application และ เลือก Ok ดังหมายเลข 12หมายเลข 13 และ หมายเลข 14

#### 9. เลือก Windows Forms Application แล้วกดปุ่ม OK



# 10. ถึงหน้านี้แสดงว่าโปรแกรม Visual Studio 2010 พร้อมใช้งานแล้ว

ee WindowsApplication1 - Microsoft Visual Studio (Administrator)	
File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help	
[] ] ● 22 田 • 月 24   X 42   34   三 2   ク・ペ・ジーム  ◆ エョックロ (Phil Debug ・) 句子 古田 火星 (Phil Fill)	
A FormLyb [Design] ×	Solution Explorer 🔹 👎 🗙
🖗 🧬 Form1.	WindowsApplication1
	My Project
	(and remaine)
	Solution Fr. Taxon Funda
	Solution Ex
	Properties • • • ×
	Form1.vb File Properties
	24
	Copy to Outp Do not copy
	Custom Tool
	Custom Tool
	File Name Form1.vb
	Build Action How the file relates to the build and
	deployment processes.
Ready	

## ิการใช้โปรแกรม Visual Studio 2010

# 3.4.1 เริ่มต้นใช้งาน Visual C# 2010

เมื่อมีการเริ่มต้นใช้งาน Visual Studio 2010 เป็นครั้งแรก จะมีขั้นตอนเพื่อเลือกใช้ Visual C#2010 เป็นภาษาหลักสำหรับเขียนโปรแกรมคอมพิวเตอร์ ดังนี้

คลิกปุ่ม Start > All Programs > Microsoft Visual Studio 2010 > Microsoft
 VisualStudio 2010 จะปรากฏหน้าต่าง Choose Default Environment Settings ขึ้นมา
 โดยแสดงดังรูปที่ 3.9

2. คลิกเลือก Visual C# Development Settings เพื่อใช้ภาษา Visual C# เป็นภาษา หลักสำหรับเขียนโปรแกรมคอมพิวเตอร์ โดยแสดงดังรูปที่ 3.9

3. คลิกปุ่ม Start Visual Studio โดยแสดงดังรูปที่ 3.9



รูปที่3.9 ไดอะล็อกบ็อกซ์ Choose Default Environment Settings

## 4. คลิกที่ New Project... โดยแสดงดังรูปที่ 3.10



5. คลิกเลือก Windows Forms Application โดยแสดงดังรูปที่ 3.11

6. ตั้งชื่อ Project ซึ่งก็คือ ชื่อโปรแกรมคอมพิวเตอร์ที่จะสร้างขึ้นมานั้นเอง โดยแสดงดัง รูปที่ 3.11

# 7. คลิกปุ่ม OK โดยแสดงดังรูปที่ 3.11



----

# 8. จากนั้นจะพบกับหน้าจอหลักของ Visual Studio 2010 โดยแสดงดังรูปที่ 3.12

File Git View Project Build Debug Term Data Tools Architecture Test Ankyre Window Help   Image: Contract Build Debug Term Data Tools Architecture Test Ankyre Window Help   Image: Contract Build Debug Term Data Tools Architecture Test Ankyre Window Help   Image: Contract Build Debug Term Data Tools Architecture Test Ankyre Window Help   Image: Contract Build Debug Term Data Tools Architecture Test Ankyre Window Help   Image: Contract Build Debug Term Data Tools Architecture Test Ankyre Window Help   Image: Contract Build Debug Term Data Tools Architecture Test Ankyre Window Help   Image: Contract Build Debug Test Ankyre Window Help   Image: Contract Build Debu	🚥 WindowsFormsApplication14 - Microsoft Visual Studio	– 0 ×
Image: Contract of the second seco	File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help	
Femlas (Design) ×     Solution Explore     • 0 ×       Femlas (Design) ×     • 0 × 0 ×       • 0 × 0 ×     • 0 ×       • 0 × 0 ×     • 0 ×       • 0 × 0 ×     • 0 ×       • 0 × 0 ×     • 0 ×       • 0 × 0 ×     • 0 ×       • 0 × 0 ×     • 0 ×       • 0 × 0 ×     • 0 ×       • 0 ×     • 0 ×       • 0 ×     • 0 ×       • 0 ×     • 0 ×       • 0 ×     • 0 ×       • 0 ×     • 0 × </td <td></td> <td></td>		
Femile (beign) ×     Solution replore     + + ×       Image: Constraint of the second	:#[변수례]∿~~~~[안재띯쌺 ~~☆☆☆]&장양안만[[D]]례례]@ @ =	
Ready	Form1 Control	<ul> <li>Solution Explorer</li> <li>Solution WindowsFormsApplic</li> <li>WindowsFormsApplication</li> <li>WindowsFormsApplication</li> <li>Properties</li> <li>References</li> <li>Form1.cs</li> <li>Program.cs</li> </ul>

# ้ 3.4.2 องค์ประกอบของ IDE

ก่อนเริ่มต้นเขียนโปรแกรมคอมพิวเตอร์ด้วยภาษา C# ควรจะมารู้จักกับเครื่องมือ สาหรับเขียนโปรแกรมคอมพิวเตอร์ Visual C# 2010 เพื่อให้สามารถใช้เครื่องมือได้อย่าง มีประสิทธิภาพ โดยแสดงดังรูปที่ 3.13

eo WindowsFormsApplication14 - Microsoft V File Edit View Project Build Debug 한국 - 한국 - 20 국 2	Ysual Studio Team Data Tools Architecture Test Analyze Window - 안 - 글리 - 집금   ▶ Debug - 1266 금 대   ∞ 같도 않 않 않 금 중 중: 음: 여: 말 문 ( 급 급) ( 급		Menu Bar	×881.	- 0 >
Toolbox     ▼ □ ×       > All Window Forms     ▲       Common Centrols     ▲       Ponter     ●       ● Button     □       ♡ Crect     TOOlbox       ■ Date     ■       ▲ LinkLabel     ●       ■ ListBox     #       # MosthClexIdex     MonthCalendar       ■ Notifyicon     ■       ■ Notifyicon     ■	Form1	Form D	esign	<ul> <li>Solution Epiloas</li> <li>Solution Vindo</li> <li>Solution Vindo&lt;</li></ul>	Toolbar wsformsApplication14 (1) pr msApplication14 besigner.cs Lesigner.cs cs v · · · · · · · · · · · · · · · · · · ·
PictureBox ProgressBar RadioButton RadioB	,			Showicon Showicon Showicon Size SizeGripStyle StartPosition Tag Test TopMost TransparencyKey UseWwitCuror Text The text associated wit	True True True True Sou, 300 Auto WindowsDefaultLocati Form False

**รูปที่** 3.13 หน้าจอแสดงองค์ประกอบหลักของ IDE

# จากรูปที่ 3.13 แสดงหน้าจอแสดงองค์ประกอบหลักของ IDE โดยมีรายละเอียด ดังนี้

## 1. เมนูบาร์ (Menu Bar)

เมนูบาร์ เป็นเมนูหลักที่รวบรวมคำสั่งควบคุมการทางานของ Visual Studio 2010 โดยจัดเป็นกลุ่มคำสั่งแยกตามประเภทการใช้งาน

## 2. ทูลบาร์ (Toolbar)

ทูลบาร์ เป็นการรวบรวมคำสั่งในเมนูบาร์บางคำสั่งที่มีการใช้งานบ่อย ๆ มาแสดงไว้ เพื่อให้สามารถเรียกใช้งานได้สะดวกขึ้น โดยสร้างเป็นปุ่มให้เรียกใช้งานได้ในคลิกเดียว

# 3. หน้าต่าง Toolbox

Toolbox เป็นหน้าต่างที่แสดงคอนโทรลและคอมโพเนนต์ต่าง ๆ เพื่อให้สะดวกในการ สร้างโปรแกรมคอมพิวเตอร์ จึงมีการจัดแบ่งคอนโทรลและคอมโพเนนต์ต่าง ๆ ออกเป็น กลุ่ม ๆ ตามลักษณะการใช้งาน

## 4. หน้าต่าง Form Designer

Form Designer เป็นหน้าต่างที่ใช้สาหรับการออกแบบหน้าตาของโปรแกรม คอมพิวเตอร์โดยการลากคอนโทรลต่าง ๆ จาก Toolbox Window มาวางบน Form ตาม ต้องการ

## 5. หน้าต่าง Solution Explorer

Solution Explorer เป็นหน้าต่างแสดงรายการของไอเท็ม (Item) ทั้งหมดที่มีอยู่ในโปร เจ็กต์ เช่น Form, Module, Component และ Class เป็นต้น

## 6. หน้าต่าง Properties Window

Properties Window เป็นหน้าต่างที่ใช้แสดงและกำหนดคุณสมบัติของชิ้นส่วน (หรือ องค์ประกอบ) ต่าง ๆ ของโปรแกรมคอมพิวเตอร์ที่กาลังจะสร้างขึ้น

## 7. หน้าต่าง Code Editor

Code Editor เป็นหน้าต่างที่ใช้สาหรับเขียนคำสั่งในภาษา C# เพื่อกำหนดการทางาน ของโปรแกรมคอมพิวเตอร์ตามที่ต้องการ





4.1 วิธีการเรียกใช้โปรแกรม Visual C# 4.2 โครงสร้างคำสั่งโปรแกรม Visual C#

4.3 องค์ประกอบพื้นฐาน Visual C#

4.4 การเขียนโปรแกรมในโหมด Console Application ด้วย Visual C#

## วิธีการเรียกใช้โปรแกรม Visual C#

4.1

์ โปรแกรม Visual C# หลังจากติดตั้งโปรแกรมเสร็จแล้ว ก็จะสามารถเข้าใช้โปรแกรมได้ ทันทีโดยมีขั้นตอนดังนี้

เมื่อเข้าสู่โปรแกรม Visual Studio 2010 จะพบหน้าต่างดังนี้







## **รูปที่ 4.2** เลือก Console Application แล้วคลิก OK

File Edit View Refactor Project Build Debug Team Data Tools Architecture Teat Analyze Window Help (1 • • • • • • • • • • • • • • • • • • •		
3・10・20 - 20 × 20 × 20 × 20 × 20 × 20 × 20 ×		
김 정말 사람 總統 출입 미지의 의사의 정 것.		
Program.cs. X	lution Explorer	- 4 ×
Consoledpplication1Program     (@Main(string[] args)     ()     class Program     class Program     ()     static void Main(string[] args)     (	Solution Explorer Team Expl	1 project)
	operties	• • •
100 % • <	21	
EnorList + 9 ×		
O D Errors 🛕 0 Warnings 🕕 0 Messages		
Description File Line Column Project		
testy to 13 of	Colé Ché	INS

# **รูปที่ 4.3** หน้าต่างโปรแกรม Visual C#

-

# จากนั้นให้ทดลองเขียนโปรแกรมตัวอย่างดังนี้ต่อไปนี้

```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello Visual Studio C# 2010");
            Console.ReadKey();
        }
    }
}
```

ผลลัพธ์ที่ได้จากการรันโปรแกรมดังนี้

Hello Visual Studio C# 2010

## โครงสร้างคำสั่งโปรแกรม Visual C#

# โปรแกรมภาษา Visual C# มีรูปแบบโครงสร้างคำสั่ง ดังต่อไปนี้

4.2



# .3 องค์ประกอบพื้นฐาน Visual C#

# ์ 4.3.1 บล็อก {...}

รูปแบบของโปรแกรม Visual C# จะใช้บล็อก {...} ในการกำหนดจุดเริ่มต้นและ จุดสิ้นสุดของการทำงานในแต่ละส่วนของโปรแกรม ซึ่งภายในแต่ละบล็อกอาจมีบล็อก ย่อย ๆ ซ้อนลงไปได้อีก ตามลักษณะของงาน เช่น

Class className	
{	
If ()	
{	
for ()	
{	
}	
}	
}	

# 4.3.2 เครื่องหมายสิ้นสุดคำสั่ง (;)

ในโปรแกรม Visual C# เราจะใช้เครื่องหมายเซมิโคลอน (;) เป็นตัวแสดงจุดสิ้นสุดของ แต่ละคำสั่ง หากเราไม่ใส่เครื่องหมายนี้เพื่อคั่นระหว่างคำสั่งแล้ว โปรแกรมจะถือว่าเป็นคำสั่ง เดียวกันไปตลอดถึงแม้ว่าจะอยู่คนละบรรทัดก็ตาม เช่น

X = 20;

Y = "xxx";

Z = x +

20;

เมื่อเครื่องหมาย ; เป็นตัวบ่งบอกว่าเป็นจุดสิ้นสุดของคำสั่ง คำสั่งต่าง ๆ จึงจะสามารถ มาอยู่ในบรรทัดเดียวกันได้ เช่น

X = 20; y = "xxx"; z = x + 20;

แต่ในการเขียนรหัสคำสั่งในลักษณะนี้ จะไม่นิยมเขียน เนื่องจากอ่านโปรแกรมได้ยาก และรหัสโปรแกรมดูไม่เป็นระเบียบ แต่อาจนำมาใช้ในบางกรณีได้

# 4.3.3 การเขียนคำอธิบายประกอบในรหัสโปรแกรม

คำอธิบาย หมายถึง การเขียนข้อความใด ๆ ที่ไม่ใช่คำสั่งแต่เขียนปะปนกันไปกับคำสั่ง อื่น ๆ เพื่ออธิบายเรื่องใดเรื่องหนึ่งเอาไว้เพื่อความเข้าใจในรหัสโปรแกรมตรงส่วนนั้น ๆ โดย ทั้งนี้เพื่อให้โปรแกรมไม่สับสนว่าส่วนใดเป็นคำสั่ง ส่วนใดเป็นแค่เพียงคำอธิบายการแทรก คำอธิบาย โดยสามารถทำได้ 2 ลักษณะดังนี้ คือ

## 1. รูปแบบ // คำอธิบาย

จะใช้สำหรับอธิบายบรรทัดเดียว โดยโปรแกรมจะถือว่าตั้งแต่สัญลักษณ์ // เป็นต้นไปจนถึงสิ้นสุดบรรทัดจะเป็นคำอธิบายทั้งหมด จะไม่นำมาพิจารณาใน การประมวลผลของโปรแกรม เช่น

// สูตรการคำนวณหาขนาดพื้นที่สี่เหลี่ยมผืนผ้า

RectangleArea=width\*length;

หรือ

RectangleArea=width\*length; //สูตรการคำนวณหาขนาดพื้นที่ สี่เหลี่ยมผืนผ้า

## 2. รูปแบบ /\* คำอธิบาย \*/

ในกรณีที่คำอธิบายโปรแกรมของเราค่อนข้าง ยาว จำเป็นต้องเขียนหลาย ๆ บรรทัด การใช้// หลาย ๆ ครั้ง จึงอาจไม่สะดวกนัก เราสามารถใช้ /\*...\*/ ครอบคำอธิบายนั้นแทนได้ ซึ่งโปรแกรมจะ ถือว่า /\*เป็นต้นไปจะเป็นคำอธิบายทั้งหมด จนกว่าจะเจอสัญลักษณ์ \*/ จึงจะถือว่าเป็นการ สิ้นสุดคำอธิบาย เช่น

/\* นี่เป็นส่วนของคำอธิบาย ที่ไม่มีผลต่อการทำงานของโปรแกรม มีไว้เพื่อช่วยให้ทำความเข้าใจในการเขียน รหัสคำสั่งได้ง่ายขึ้น \*/ 🖡 🔪 การเขียนโปรแกรมในโหมด Console Application ด้วย Visual C#

4.4.1 คำสั่งการแสดงข้อมูลออกทางจอภาพ แบ่งออกเป็น 2 คำสั่ง ดังต่อไปนี้ 1. คำสั่ง Write() รูปแบบคำสั่ง Console.Write(อักขระ/สตริง/ตัวเลข/ตัวแปร/นิพจน์/เมธอด); คำอธิบาย การแสดงข้อมูลสามารถแสดงได้หลายรูปแบบดังต่อไปนี้

(1) อักขระ หมายถึง ตัวอักขระทั่ว ๆ ไป โดยตัวอักขระจะต้องอยู่ภายใต้เครื่องหมาย '...'และในเครื่องหมายนี้จะมีอักขระได้เพียง 1 ตัวอักขระเท่านั้น เช่น Console.Write('A');

(2) สตริง หมายถึง ประโยคข้อความต่าง ๆ โดยสตริงจะต้องอยู่ภายใต้เครื่องหมาย "…"และจะมีกี่ตัวอักขระก็ได้ เช่น

Console.Write("Hello");

(4) ตัวแปร หมายถึง ตัวแปรทุกตัวในโปรแกรม ไม่ว่าจะเป็นตัวแปรชนิดสตริง ตัวเลขหรือบูลีน เช่น

int age = 50;

Console.Write(age);

(5) นิพจน์ หมายถึง นิพจน์ต่าง ๆ เช่น นิพจน์ทางคณิตศาสตร์ นิพจน์เปรียบเทียบ เป็นต้น เช่น

int age = 50;

Console.Write(age/2); //นิพจน์ทางคณิตศาสตร์ Console.Write(age>30); //นิพจน์เปรียบเทียบ

(6) **เมธอด** หมายถึง ทั้งเมธอดสำเร็จและเมธอดที่เราสร้างขึ้นที่มีการส่งค่ากลับ เช่น

Console.Write(Math.sqrt(9));

# คำสั่ง WriteLine() รูปแบบคำสั่ง

## Console.WriteLine(อักขระ/สตริง/ตัวเลข/ตัวแปร/นิพจน์/เมธอด);

จะเห็นว่าคำสั่ง WriteLine() สามารถแสดงข้อมูลได้เหมือนกับคำสั่ง Console.Write() แต่แตกต่างกันที่ผลลัพธ์ในการแสดงผล ดังตัวอย่างต่อไปนี้



# 4.4.2 คำสั่งอ่านข้อมูลจากคีย์บอร์ด

เป็นคำสั่งสำหรับอ่านหรือรับข้อมูลจากคีย์บอร์ด ณ จุดที่รับคำสั่งหรือจุดที่ เคอร์เซอร์รออยู่ มี2 รูปแบบคือ

1. Console.Read() เป็นการอ่านค่าข้อมูลในรูปแบบของ Integer (int) ดังตัวอย่าง

using System; using System.Collections.Generic; using System.Text; namespace ConsoleApplication1 class Program static void Main(string[] args) int i = Console.Read();

## 2. Console.ReadLine() เป็นการอ่านค่าข้อมูลในรูปแบบของ string ดังตัวอย่าง

```
using System;
using System.Collections.Generic;
using System.Text;
namespace ConsoleApplication1
{
  class Program
     {
        static void Main(string[] args)
           string str = Console.ReadLine();
         }
```



# ตัวอย่างที่ 1

using System;

using System.Collections.Generic;

using System.Text;

namespace ConsoleApplication1

{ class Program

static void Main(string[] args)

{

Console.Read();

# ผลลัพธ์โปรแกรม

Empty program \_

# ตัวอย่างที่ 2

using System; using System.Collections.Generic; using System.Text; namespace ConsoleApplication1 { class Program { static void Main(string[] args) Console.Write("Programming in C# is easy."); Console.Read();

## ผลลัพธ์โปรแกรม

Programming in C# is easy.\_





5.1 การออกแบบโปรแกรม เชิงวัตถุเบื้องต้นเพื่อประยุกต์ ในงานธุรกิจ โดยใช้คำสั่งรับ ข้อมูลผ่านทางคีย์บอร์ด

(Keyboard)





การออกแบบโปรแกรมเชิงวัตถุเบื้องต้นเพื่อประยุกต์ในงานธุรกิจ โดยใช้คำสั่งรับข้อมูล ผ่านทางคีย์บอร์ด (Keyboard)

# 5.1.1 ตัวอย่างที่ 1 โปรแกรมการหาพื้นที่สี่เหลี่ยมผืนผ้า ที่กำหนดค่า ให้กับตัวแปรแบบตายตัว

# <mark>ตัวอย่างที่</mark> 1 จงเขียนโปรแกรมการหาพื้นที่สี่เหลี่ยมผืนผ้า ที่กำหนดค่าให้กับตัวแปรแบบตายตัว

## โค้ตโปรแกรม

แบเ

	using System;	
	using System.Collections.Generic;	
	using System.Linq;	
	using System.Text;	
	namespace ConsoleApplication3	
	{	
	class Program	
	{	
	static void Main(string[] args)	
	{	
	//ตัวอย่างโปรแกรมหาพื้นที่สี่เหลี่ยมผืนผ้าที่กำหนดค่าให้กับตัวแปรที่นำเข้า (Input)	
มตายตัว		
	int width = 5, length = 20;	
	int RectangleArea = width * length;	



#### ผลลัพธ์โปรแกรม

Rectangle Area = 100

#### อธิบายโปรแกรม

int width = 5, length = 20;

int RectangleArea = width \* length;

width เป็นตัวแปรที่กำหนดให้เป็นชนิดแบบ int เลขจำนวนเต็ม จะเก็บค่าความกว้าง โดย กำหนดตายตัวให้มีค่าเท่ากับ 5

length เป็นตัวแปรที่กำหนดให้เป็นชนิดแบบ int เลขจำนวนเต็ม จะเก็บค่าความยาว โดย กำหนดตายตัวให้มีค่าเท่ากับ 20

RectangleArea เป็นตัวแปรที่กำหนดให้เป็นชนิดแบบ int เลขจำนวนเต็ม จะเก็บพื้นที่ สี่เหลี่ยมผืนผ้า

โดยใช้คำสั่ง WriteLine ในการแสดงผลลัพธ์ของพื้นที่สี่เหลี่ยมผืนผ้า ซึ่งจะเก็บผลลัพธ์ไว้ใน ตัวแปร RectangleArea

# 5.1.2 ตัวอย่างที่ 2 โปรแกรมการหาพื้นที่สี่เหลี่ยมผืนผ้า ที่รับข้อมูล ผ่านทางแป้นพิมพ์(Keyboard)

# **ตัวอย่างที่ 2** จงเขียนโปรแกรมการหาพื้นที่สี่เหลี่ยมผืนผ้า ที่รับข้อมูลผ่านทางแป้นพิมพ์

## โค้ดโปรแกรม

using System; using System.Collections.Generic; using System.Ling; using System.Text; namespace ConsoleApplication4 class Program static void Main(string[] args) //ตัวอย่างโปรแกรมหาพื้นที่สี่เหลี่ยมผืนผ้าที่รับข้อมูลผ่านทางแป้นพิมพ์ (Keyborad) int width, length, RectangleArea; Console.Write("Please width Enter ="); width = int.Parse(Console.ReadLine()); Console.Write("Please length Enter ="); length = int.Parse(Console.ReadLine()); RectangleArea = width \* length; Console.WriteLine("RectangleArea = {0}", RectangleArea); Console.ReadKey();

### ผลลัพธ์โปรแกรม

Please width Enter = 10 Please length Enter = 20 RectangleArea = 200

## อธิบายโปรแกรม

Int width, length, RectangleArea; width เป็นตัวแปรที่กำหนดให้เป็นชนิดแบบ int เลขจำนวนเต็ม จะเก็บค่าความกว้าง length เป็นตัวแปรที่กำหนดให้เป็นชนิดแบบ int เลขจำนวนเต็ม จะเก็บค่าความยาว RectangleArea เป็นตัวแปรที่กำหนดให้เป็นชนิดแบบ int เลขจำนวนเต็ม จะเก็บพื้นที่ สี่เหลี่ยมผืนผ้า

โดยใช้คำสั่ง ReadLine เพื่อรับข้อมูลผ่านทางคีย์บอร์ด (Keyboard)

width = int.Parse(Console.ReadLine());

```
length = int.Parse(Console.ReadLine());
```

และใช้คำสัง WriteLine ในการแสดงผลลัพธ์ของพื้นที่สี่เหลี่ยมผืนผ้า ซึ่งจะเก็บผลลัพธ์ไว้ ในตัวแปร RectangleArea

RectangleArea = width \* length;

Console.WriteLine("RectangleArea = {0}", RectangleArea);

# 5.1.3 ตัวอย่างที่ 3 โปรแกรมการคำนวณหาใบเสร็จรับเงิน

**ตัวอย่างที่** 3 ให้เขียนโปรแกรมเพื่อคำนวณหาใบเสร็จรับเงิน โดยมีรายละเอียดของข้อมูล ดังนี้ รหัสสินค้า ชื่อสินค้า จำนวนสินค้า ราคาสินค้า ราคารวม

#### โค้ดโปรแกรม

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication6
{
class Program
{
static void Main(string[] args)
{
//ตัวอย่างโปรแกรมหาการคำนวณหาใบเสร็จรับเงิน ที่รับข้อมูลผ่านทางแป้นพิมพ์ (Keyborad)
int code,quantity;
string name;
double price, total;
Console.WriteLine("====================================
Console.WriteLine(" Program Receipt ");
Console.WriteLine("====================================
Console.Write("Please code Enter = ");

	code = int.Parse(Console.ReadLine());
	Console.Write("Please name Enter = ");
	name = Console.ReadLine();
	Console.Write("Please quantity Enter = ");
	quantity = int.Parse(Console.ReadLine());
	Console.Write("Please price Enter = ");
	price = double.Parse(Console.ReadLine());
	total = quantity * price;
	Console.WriteLine("====================================
	Console.WriteLine("total = {0:##,###.00}", total);
	Console.WriteLine("========"");
	Console.ReadKey();
	}
}	

#### ผลลัพธ์ของโปรแกรม

Promgram Receipt

Please code Enter = 1001

Please name Enter = computer

Please quantity Enter = 1

Please price Enter = 15000

total = 15,000.00
### อธิบายโปรแกรม

int code, quantity;

string name;

double price, total;

code เป็นตัวแปรที่ใช้แทนการเก็บค่ารหัสสินค้า โดยกำหนดให้เป็นชนิดแบบ int เลข จำนวนเต็ม

quantity เป็นตัวแปรที่ใช้แทนการเก็บค่าจำนวนสินค้า โดยกำหนดให้เป็นชนิดแบบ int เลขจำนวนเต็ม

name เป็นตัวแปรที่ใช้แทนการเก็บค่าชื่อสินค้า โดยกำหนดให้เป็นชนิดแบบ string รับ เป็นตัวอักขระ

price เป็นตัวแปรที่ใช้แทนการเก็บค่าราคาสินค้า โดยกำหนดให้เป็นชนิดแบบ double เลขจำนวนทศนิยม

# 5.1.4 ตัวอย่างที่ 4 โปรแกรมการคำนวณหาส่วนลดสินค้า

ตัวอย่างที่ 4 ให้เขียนโปรแกรม เพื่อคำนวณหาใบเสร็จรับเงิน โดยมี รายละเอียดของข้อมูลดังนี้รหัสสินค้า ชื่อสินค้า จำนวนสินค้า ราคาสินค้า ราคารวม ส่วนลดสินค้า 10% ราคา สุทธิ

### โค้ดโปรแกรม

using System; using System.Collections.Generic; using System.Linq; using System.Text; namespace ConsoleApplication7 { class Program

static void Main(string] args)

//ตัวอย่างโปรแกรมหาการคำนวณหาส่วนลดสินค้าโดยคิดส่วนลด 10% ที่รับข้อมูล ผ่านทางแป้นพิมพ์ (Keyborad)

int code, quantity;	
string name;	
double price, total,discount,nettotal;	
Console.WriteLine("====================================	===");
Console.WriteLine(" Program Receipt	")
Console.WriteLine("====================================	===");
Console.Write("Please code Enter = ");	
code = int.Parse(Console.ReadLine());	
Console.Write("Please name Enter = ");	
name = Console.ReadLine();	
Console.Write("Please quantity Enter = ");	
quantity = int.Parse(Console.ReadLine());	
Console.Write("Please price Enter = ");	
price = double.Parse(Console.ReadLine());	
total = quantity * price;	
discount = total * 0.10;	



WORLD'S BEST



Console.WriteLine("========================"); Console.WriteLine("total = {0:##,###.00}", total);

Console.WriteLine("discount = {0:##,###.00}", discount);

Console.WriteLine("nettotal = {0:##,###.00}", nettotal);

### ผลลัพธ์ของโปรแกรม

#### Promgram Receipt

Please code Enter = 1002

Please name Enter = computer

Please quantity Enter = 2

Please price Enter = 25000

total = 50,000. 00 discount = 5,000. 00

nettotal = 45,000.00

### อธิบายโปรแกรม

int code, quantity;

string name;

double price, total,discount,nettotal; code เป็นตัวแปรที่ใช้แทนการเก็บค่ารหัสสินค้า โดย กำหนดให้เป็นชนิดแบบ int เลขจำนวนเต็ม quantity เป็นตัวแปรที่ใช้แทนการเก็บค่าจำนวนสินค้า โดย กำหนดให้เป็นชนิดแบบ int เลขจำนวนเต็ม name เป็นตัวแปรที่ใช้แทนการเก็บค่าชื่อสินค้า โดย กำหนดให้เป็นชนิดแบบ string รับเป็นตัวอักขระ price เป็นตัวแปรที่ใช้แทนการเก็บค่าราคาสินค้า โดย กำหนดให้เป็นชนิดแบบ double เลขจำนวนทศนิยม

# 5.1.5 ตัวอย่างที่ 5 โปรแกรมการคำนวณหาค่านายหน้า (Commission)

ตัวอย่างที่ 5 ให้เขียนโปรแกรม เพื่อคำนวณหาค่านายหน้า โดยคิดค่า คอมมิชชั่น 5% จากยอดขายรวม สินค้า 3 ชนิด โดยมีรายละเอียดข้อมูล ดังนี้ รหัสประจำตัวพนักงาน ชื่อ พนักงาน ยอดขายสินค้าชนิดที่ 1 ยอดขายสินค้าชนิดที่ 2 ยอดขาย สินค้าชนิดที่ 3 ยอดขายรวมสินค้า 3 ชนิด ค่าคอมมิชชั่น ยอดขายสุทธิ

### โค้ดโปรแกรม

using S	System;		
using S	System.Collections.Generic;		
using S	System.Linq;		
using	System.Text;		
names	space ConsoleApplication8		
{			
cla	ss Program		
{			
	static void Main(string[] args)		
÷	{		
	//ตัวอย่างโปรแกรมหาการคำ	นวณหาค่านายหน้า 5% จากยอดขายรวมสิน	ค้า 3 ชนิด
ไร้บข้อมูลผ่านทางแป้เ	นพิมพ์ (Keyborad)		
	int EmployeeID;		
	string name;		
	double sale1,sale2,sale3,	total,commis,netsale;	
	Console.WriteLine("====		=====");
	Console.WriteLine("	Program Commission	");
	Console.WriteLine("====		=====");
	Console.Write("Please Em	ployeeID Enter = ");	
	EmployeeID = int.Parse(C	onsole.ReadLine());	
	Console.Write("Please name	me Enter =   ");	
	name = Console.ReadLine	e();	
	Console.Write("Please Pro	oductsale1 Enter = ");	
	sale1 = double.Parse(Co	nsole.ReadLine());	
	Console.Write("Please Pro	oductsale2 Enter = ");	



}

#### ผลลัพธ์โปรแกรม

Promgram Commission

Please EmployeeID Enter = 12345

Please name Enter = warapron

Please Productsale1 Enter = 1000

Please Productsale2 Enter = 2000

Please Productsale3 Enter = 3000

total = 6,000. 00 commission = 300. 00

netsale = 5,700.00

### อธิบายโปรแกรม

int EmployeeID;

string name;

double sale1,sale2,sale3,total,commis,netsale; EmployeeID เป็นตัวแปรที่ใช้แทนการเก็บค่ารหัสประจำตัวพนักงาน โดย กำหนดให้เป็นชนิดแบบ int เลขจำนวนเต็ม name เป็นตัวแปรที่ใช้แทนการเก็บค่าชื่อพนักงาน โดยกำหนดให้เป็นชนิดแบบ string รับเป็นตัวอักขระ sale1 เป็นตัวแปรที่ใช้แทนการเก็บค่ายอดขายสินค้าชนิดที่ 1 โดยกำหนดให้เป็น ชนิดแบบdouble เลขจำนวนทศนิยม sale2 เป็นตัวแปรที่ใช้แทนการเก็บค่ายอดขายสินค้าชนิดที่ 2 โดยกำหนดให้เป็น ชนิดแบบdouble เลขจำนวนทศนิยม sale3 เป็นตัวแปรที่ใช้แทนการเก็บค่ายอดขายสินค้าชนิดที่ 3 โดยกำหนดให้เป็น ชนิดแบบdouble เลขจำนวนทศนิยม







6.1 คำสั้ง lf

6.2 การออกแบบโปรแกรม
 เชิงวัตถุเบื้องต้นเพื่อประยุกต์
 ในงานธุรกิจ โดยใช้คำสั่ง
 โครงสร้าง If และ If...else





คำสั่ง if เป็นโครงสร้างที่ใช้ควบคุมการทำงานของคำสั่งอื่น ๆ ภายใต้เงื่อนไข (condition) ที่กำหนดการใช้งานจะมีสองรูปแบบดังนี้

# 6.1.1 รูปแบบที่ 1 lf

จากการรูปแบบการใช้งานด้านล่าง คำสั่ง statement จะถูกเรียกทำงานก็ต่อเมื่อ นิพจน์ทางตรรกศาสตร์ที่กำหนดเป็น condition มีค่าเป็นจริง

if (condition)

statement; // executed if the condition is true

เนื่องจากโครงสร้างข้างต้นจะอนุญาตให้เรากำหนดเงื่อนไขให้กับคำสั่งเพียงคำสั่ง เดียวเท่านั้นอย่างไรก็ตาม หากมีคำสั่งมากกว่าหนึ่งภายใต้เงื่อนไขเดียวกัน คำสั่งเหล่านี้ สามารถถูกจัดกลุ่มให้เป็นเสมือนคำสั่งเดียวได้โดยการครอบคำสั่งทั้งหมดด้วยวงเล็บปีก กา ({...})

if (condition) {

statement1; // executed if the condition is true statement2; // executed if the condition is true statement3; // executed if the condition is true

# 6.1.2 รูปแบบที่ 2 lf...else

คำสั่ง statement1 จะถูกเรียกทำงานเมื่อนิพจน์ในตำแหน่ง condition มีค่าเป็น จริง หากนิพจน์ดังกล่าวมีค่าเป็นเท็จ คำสั่ง statement2 จะถูกเรียกทำงานแทน

if (condition)

statement1; //executed if the condition is true

else

statement2; //executed if the condition is false

ดังนั้น เราจึงสามารถใช้งานคำสั่ง if...else ร่วมกับวงเล็บปีกกาหากมีคำสั่งที่ต้องการ ให้ทำงานภายใต้เงื่อนไขมากกว่าหนึ่ง

if (condition) {

statementT1; //executed if the condition is true statementT2; //executed if the condition is true

}

else {

statementF1; //executed if the condition is false

statementF2; //executed if the condition is false



การออกแบบโปรแกรมเชิงวัตถุเบื้องต้นเพื่อประยุกต์ในงานธุรกิจ โดยใช้คำสั่งโครงสร้าง If และ If...else

# 6.2.1 ตัวอย่างที่ 1

บริษัท วีระ อิเล็กทรอนิกส์ จำกัด ต้องการคำนวณหาส่วนลดของสินค้า ของบริษัท โดยมีเงื่อนไขในการคำนวณหาส่วนลดดังนี้ ถ้าซื้อน้อยกว่า 500 บาท ไม่ลดราคา (หมายถึงน้อยกว่าหรือเท่ากับ 499) ถ้าซื้อระหว่าง 500–2,000 บาท ให้ส่วนลด 3% ถ้าซื้อระหว่าง 2,001–4,000 บาท ให้ส่วนลด 5% ถ้าซื้อระหว่าง 4,001–6,000 บาท ให้ส่วนลด 7% ถ้าซื้อมากกว่า 6,000 บาทขึ้นไป ให้ส่วนลด 10% รายละเอียดข้อมูลประกอบด้วย รหัสสินค้า, รายการสินค้า, จำนวนสินค้า, ราคาสินค้า, ราคารวม, ส่วนลดสินค้า, ราคาหลังหักส่วนลด

### โค้ดโปรแกรม

using System; using System.Collections.Generic; using System.Linq; using System.Text; namespace ConsoleApplication9 class Program static void Main(string[] args) int code, quantity; string name; double price, total, discount, nettotal; Console.WriteLine(" Program Receipt "); Console.Write("Please code Enter = ");



code = int.Parse(Console.ReadLine()); Console.Write("Please name Enter = "); name = Console.ReadLine(); Console.Write("Please quantity Enter = "); quantity = int.Parse(Console.ReadLine()); Console.Write("Please price Enter = "); price = double.Parse(Console.ReadLine()); total = quantity \* price; if (total < 500)discount = total \* 0; else if (total < 2001) discount = total \* 0.03; else if (total < 4001) discount = total \* 0.05; else if (total < 6001) discount = total \* 0.07; else discount = total \* 0.10;

nettotal = total - discount;

Console.WriteLine("total =  $\{0:\#\#, \#\#\#, 00\}$ ", total);

Console.WriteLine("discount = {0:##,###.00}", discount);

Console.WriteLine("nettotal = {0:##,###.00}", nettotal);

="):

## ผลลัพธ์โปรแกรม

Promgram Receipt

Please code Enter = 1001

Please name Enter = computer

Please quantity Enter = 1

Please price Enter = 20000

total = 20,000.00

discount = 2,000.00

nettotal = 18,000.00

# อธิบายโปรแกรม

```
if (total < 500)
   discount = total * 0;
else if (total < 2001)
   discount = total * 0.03;
else if (total < 4001)
   discount = total * 0.05;
else if (total < 6001)
   discount = total * 0.07;
else discount = total * 0.10;
```

0 \*0 .

# 6.2.2 ตัวอย่างที่ 2

บริษัท ขนิษฐา คอมพิวเตอร์ จำกัด ต้องการคำนวณหาค่าแรง และค่าแรง ล่วงเวลา รายละเอียดของข้อมูลมีดังนี้ รหัสประจำตัวพนักงาน ชื่อพนักงาน จำนวน ชั่วโมง อัตราค่าแรง ค่าแรง ค่าแรงล่วงเวลาค่าแรงสุทธิ

โดยมีเงื่อนไขในการคิดค่าแรงดังนี้ ถ้าจำนวนชั่วโมงน้อยกว่าหรือเท่ากับ 8 ชั่วโมง คิด เป็นค่าแรงปกติ และไม่มีค่าแรงล่วงเวลา ถ้าจำนวนชั่วโมงเกิน 8 ชั่วโมง คิดเป็นค่าแรง ล่วงเวลา โดยให้คิดอัตราค่าแรงล่วงเวลาเป็นครึ่งหนึ่งของค่าแรงปกติ

code	เป็นตัวแปรที่ใช้เก็บรหัสประจำตัวพนักงาน
name	เป็นตัวแปรที่ใช้เก็บชื่อพนักงาน
hour	เป็นตัวแปรที่ใช้เก็บจำนวนชั่วโมง
rate	เป็นตัวแปรที่ใช้เก็บอัตราค่าแรง
salary	เป็นตัวแปรที่ใช้เก็บค่าแรงปกติ
overtime	เป็นตัวแปรที่ใช้เก็บค่าแรงล่วงเวลา
nettotal	เป็นตัวแปรที่ใช้เก็บค่าแรงสุทธิ

### โค้ดโปรแกรม

using System; using System.Collections.Generic; using System.Linq; using System.Text; namespace ConsoleApplication10

#### class Program

```
static void Main(string[] args)
{
    int code,hour;
    string name;
```

double rate, salary, overtime, nettotal;

Console.WriteLine(" Program Salary ");

=="):

Console.Write("Please code Enter = ");

```
code = int.Parse(Console.ReadLine());
Console.Write("Please name Enter = ");
name = Console.ReadLine();
Console.Write("Please hour Enter = ");
hour = int.Parse(Console.ReadLine());
Console.Write("Please rate Enter = ");
rate = double.Parse(Console.ReadLine());
If (hour \leq 8)
  salary = rate * hour;
  overtime = 0;
  nettotal = salary + overtime;
  Console.WriteLine("salary = \{0:\#\#,\#\#\#,00\}", salary);
  Console.WriteLine("overtime = {0:##,###.00}", overtime);
  Console.WriteLine("nettotal = {0:##,###.00}", nettotal);
```

else if (hour > 8)

Console.ReadKey();

## **ผลลัพธ์โปรแกรม** (ทำงานไม่เกิน 8 ชั่วโมง)

Promgram Salary

Please code Enter = 1234

Please name Enter = waraporn

Please hour Enter = 8

Please rate Enter = 200

salary = 1,600.00

overtime = .00

nettotal = 1,600.00

#### ผลลัพธ์โปรแกรม

(ทำงาน 8 ชั่วโมง ส่วนที่เกิน 8 ชั่วโมง คิดเป็นค่าแรงล่วงเวลา overtime โดยคิดเป็นครึ่งหนึ่ง ของค่าแรงปกติ)

Promgram Salary

Please code Enter = 12345 Please name Enter = waraporn Please hour Enter = 9 Please rate Enter = 200 salary = 1,600.00 overtime = 100.00 nettotal = 1,700.00

# ้ 6.2.3 ตัวอย่างที่ 3

ให้คำนวณหาค่าเช่าชั่วโมง Internet Café แห่งหนึ่ง โดยมีรายละเอียดการคิด ค่าบริการดังนี้

คำนวณหาค่าเช่าชั่วโมงการเล่นของลูกค้าแต่ละคน โดยมีหลักเกณฑ์ดังนี้ ถ้า ชั่วโมงการเล่น 0–2 ชั่วโมง คิดค่าเช่าชั่วโมงละ 20 บาท ถ้า ชั่วโมงการเล่น 3–5 ชั่วโมง คิดค่าเช่าชั่วโมงละ 15 บาท ถ้า ชั่วโมงการเล่น มากกว่า 5 ชั่วโมงขึ้นไป คิดค่าเช่าชั่วโมงละ 10 บาท

code
name
hour
rate
total

เป็นตัวแปรที่ใช้เก็บรหัสลูกค้า เป็นตัวแปรที่ใช้เก็บชื่อลูกค้า เป็นตัวแปรที่ใช้เก็บจำนวนชั่วโมง เป็นตัวแปรที่ใช้เก็บอัตราค่า Internet เป็นตัวแปรที่ใช้เก็บค่า Internet

### โค้ดโปรแกรม

using System; using System.Collections.Generic; using System.Linq; using System.Text; namespace ConsoleApplication11 {

```
class Program
```

```
{
```

static void Main(string[] args)

```
{
```

int code, hour;

string name;

double rate, total;

Console.WriteLine(" Program Internet Cafe ");

Console.WriteLine("========================");

```
Console.Write("Please code Enter = ");
code = int.Parse(Console.ReadLine());
Console.Write("Please name Enter = ");
name = Console.ReadLine();
Console.Write("Please hour Enter = ");
hour = int.Parse(Console.ReadLine());
if (hour \leq 2)
  rate = 20;
  total = hour * rate;
  Console.WriteLine("rate = {0:##,###.00}", rate);
  Console.WriteLine("total = \{0:\#\#, \#\#\#.00\}", total);
  Console.WriteLine("========================");
else if (hour \leq 5)
  rate = 15;
  total = hour * rate;
  Console.WriteLine("rate = {0:##,###.00}", rate);
  Console.WriteLine("total = \{0:\#\#, \#\#\#, 00\}", total);
```

```
else if (hour > 5)
   rate = 10;
   total = hour * rate;
   Console.WriteLine("rate = {0:##,###.00}", rate);
   Console.WriteLine("total = {0:##,###.00}", total);
   Console.ReadKey();
```

# **ผลลัพธ์โปรแกรม** เมื่อใช้ Internet 2 ชั่วโมง คิดชั่วโมงละ 20 บาท

#### Promgram internrt Cafe

Please code Enter = 12345 Please name Enter = waraporn Please hour Enter = 2 rate = 20.00 total = 40.00

# **ผลลัพธ์โปรแกรม** เมื่อใช้ Internet มากกว่า 2 ชั่วโมง แต่ไม่เกิน 5 ชั่วโมง คิดชั่วโมงละ 15 บาท

Promgram internrt Cafe

Please code Enter = 12345

Please name Enter = waraporn

Please hour Enter = 5

rate = 15.00

total = 75.00





# 7.1 คำสั่ง Switch case

# 7.2 ตัวอย่างโปรแกรม



# 7.1 🎽 คำสั่ง Switch case

ถึงแม้ว่าการใช้คำสั่ง if เพื่อจัดการเงื่อนไขหลาย ๆ เงื่อนไขในคราวเดียวกันจะสามารถทำได้โดย อาศัยโครงสร้าง if และ if...else แบบหลายชั้น ภาษา Visual C# ยังได้เตรียมคำสั่ง Switch...case เพื่อใช้ในการจัดการเงื่อนไขหลายเงื่อนไขโดยเฉพาะการใช้งานคำสั่ง Switch...case อยู่ในรูปแบบ ดังนี้

#### switch (expression)

{

case constant-expression-1:

statements;

break;

case constant-expression-2:

statements;

break;

case constant-expression-3:

statements;

break;

default:

statements;

break;

ภาษา Visual C# ยอมให้นิพจน์ที่ใช้ในตำแหน่ง expression เป็นนิพจน์ แบบจำนวนเต็ม (integer)แบบอักขระ (char) หรือแบบข้อความ (string) เท่านั้น หลังจากที่ค่าของ expression ถูกตรวจสอบโปรแกรมจะกระโดดไป ทำงาน ณ คำสั่ง case ที่ระบุค่าของ constant-expression ไว้ตรงกับค่าของ expression ที่ประเมินได้ คำสั่งต่าง ๆ ที่อยู่ถัดจากคำสั่ง case นั้น ๆ จะถูก เรียกใช้งานตามลำดับไปเรื่อย ๆ จนกว่าโปรแกรมจะพบคำสั่ง break นั้น หมายถึงจะมีผลทำให้โปรแกรมหยุดการทำงานภายในคำสั่ง switch...case ้นั้นและกระโดดไปยังคำสั่งที่ต่อไปนอกโครงสร้าง หากไม่พบ constantexpression ใดที่มีค่าตรงกับ expressionโปรแกรมจะกระโดดไป ้ยังจุดที่มีการระบุด้วยคำสั่ง default
### 7.2 🎽 ตัวอย่างโปรแกรม

### โปรแกรมต่อไปนี้แสดงแต้มคะแนนตามระดับคะแนน (A, B, C, D, F) ที่ป้อนโดยผู้ใช้

ระดับคะแนน (grade)	แต้มระดับคะแนน (grade point)
A	4.0
В	3.0
С	2.0
D	1.0
F	0.0

using System; using System.Collections.Generic; using System.Linq; using System.Text; namespace ConsoleApplication12 { class Program { static void Main(string[] args) { string grade; Console.Write("Please input your grade: "); grade = Console.ReadLine(); switch (grade)

#### case "A":

case "a": Console.WriteLine("Your point is 4.0."); break;

case "B":

case "b": Console.WriteLine("Your point is 3.0."); break; case "C":

case "c": Console.WriteLine("Your point is 2.0."); break; case "D":

case "d": Console.WriteLine("Your point is 1.0."); break; case "F":

case "f": Console.WriteLine("Your point is 0.0."); break; default: Console.WriteLine("Invalid input!!"); break;

Console.ReadKey();

#### ผลลัพธ์โปรแกรม

Please input your grade: A
Your point is 4.0.
Please input your grade: B
Your point is 3.0.
Please input your grade: e
Invalid input!!

7

แม้ว่าโปรแกรมข้างต้นจะทำงานได้อย่างถูกต้อง โปรแกรมดังกล่าวยังค่อนข้างยาวอีก ทั้งโปรแกรมยังมีคำสั่งที่ถูกใช้ซ้ำ ๆ กันอยู่หลายแห่งเนื่องจากการป้อนระดับคะแนนด้วย ตัวอักษรตัวใหญ่และตัวเล็กจะให้ผลเหมือนกัน ภาษา C# อนุญาตให้คำสั่ง case หลาย ๆ คำสั่งควบคุมชุดคำสั่งร่วมกันได้ดังตัวอย่าง

sing System;
lass GradePoint {
static void Main() {
string grade;
Console.Write("Please input your grade: ");
grade = Console.ReadLine();
switch (grade) {
case "A" :
case "a" : Console.WriteLine("Your point is 4.0."); break;
case "B" :
case "b" : Console.WriteLine("Your point is 3.0."); break;
case "C" :
case "c" : Console.WriteLine("Your point is 2.0.");            break;
case "D" :
case "d" : Console.WriteLine("Your point is 1.0."); break;
case "F" :
case "f" : Console.WriteLine("Your point is 0.0.");            break;
default: Console.WriteLine("Invalid input!!");
}
}

โปรแกรมนี้ยังสามารถทำให้สั้นลงได้อีก โดยการใช้ตัวแปรเสริมเพิ่มอีกตัวเพื่อเก็บแต้มระดับ คะแนนแล้วจึงนำค่าของตัวแปรพิมพ์ออกทางหน้าจอโดยใช้คำสั่ง Console.WriteLine ในคราว เดียวก่อนจบโปรแกรมดังตัวอย่าง

> using System; class GradePoint { static void Main() { string grade; double point = -1; Console.Write("Please input your grade: "); grade = Console.ReadLine(); switch (grade) { case "A" : case "a" : point = 4.0; break; case "B" : case "b" : point = 3.0; break; case "C" : case "c" : point = 2.0; break; case "D" : case "d" : point = 1.0; break; case "F" : case "f" : point = 0.0; break; default: Console.WriteLine("Invalid input!!"); break; if (point >= 0) Console.WriteLine("Your point is {0:f1}.", point);





## 8.1 คำสั่ง While loop

## 8.2 ตัวอย่างโปรแกรม

8.1 🎾 คำสั่ง While loop

โครงสร้าง While loop เป็นโครงสร้างแบบง่ายที่สุดที่สามารถนำมาใช้เขียน โปรแกรมเพื่อวนทำคำสั่ง(หรือกลุ่มของคำสั่ง) ซ้ำหลาย ๆ รอบ รูปแบบ การใช้ งาน While loop เป็นดังนี้

while (condition)

statement;

เช่นเดียวกับโครงสร้าง if เราสามารถกำหนดการวนซ้ำให้กับกลุ่มของคำสั่ง ได้โดยใช้วงเล็บปีกกา ({...})

while (condition) {

statement1;

statement2;

:

statementN;

ตัวอย่างโปรแกรม

8.2



### ผลลัพธ์ของโปรแกรม

Please Enter N: 5

using System.Collections.Generic;

using System.Linq;

using System;

using System.Text;

namespace ConsoleApplication14

#### class Program

{

static void Main(string[] args)
{
 int i, N;
 Console.Write("Please Enter N: ");
 N = int.Parse(Console.ReadLine());
 i = 1;
 while (i <= N)
 {
 Console.WriteLine(i);
 i++;
 }
 Console.ReadKey();
</pre>

#### อธิบายโปรแกรม

กำหนดตัวแปร i,N มีชนิดเป็น int (เลขจำนวนเต็ม) ใช้คำสั่ง Console.Write เพื่อพิมพ์ข้อความ อธิบาย Please Enter N ใช้คำสั่ง Console.ReadLine()); เพื่ออ่านค่า N กำหนดให้ตัวแปร i มีค่าเท่ากับ 1 ใช้คำสั่ง while เช็คว่าค่า i น้อยกว่าหรือเท่ากับ ค่า N ถ้าน้อยกว่าก็จะพิมพ์ค่า I ออกมา ใน คำสั่ง Console.WriteLine(i); และบวกค่า i เพิ่มทีละ 1 ในคำสั่ง i++; และวนกลับไปทำซ้ำที่ สั่ง while(i<=N) เช่น ถ้าเราป้อนเลข 5 ให้กับค่า N ดังผลลัพธ์ เมื่อค่า i มีค่าเท่ากับ 5 แล้วก็จะออกจากการวน

ลูป While ทันที ดังตัวอย่างผลลัพธ์ของ โปรแกรมข้างล่างนี้ **ตัวอย่างที่ 2** เราสามารถเขียนโปรแกรมเพื่อคำนวณหาผลรวมทั้งแต่ 1 ถึง N ใด ๆ โดยแก้ไขโปรแกรมในตัวอย่างที่ 1 เพียงเล็กน้อยเท่านั้นดังต่อไปนี้

```
using System;
using System.Collections.Generic;
using System.Ling;
using System.Text;
namespace ConsoleApplication15
  class Program
     static void Main(string[] args)
         int i, N,sum;
         Console.Write("Please Enter N: "):
         N = int.Parse(Console.ReadLine());
         i = 1; sum = 0;
         while (i <= N)
           Console.WriteLine(i);
           sum = sum + i; i++;
         Console.WriteLine("Sum from 1 to \{0\} = \{1\}", N, sum);
         Console.ReadKey();
```

#### ผลลัพธ์ของโปรแกรม

บวกเลข 1 ถึง 4 แล้วหา ผลบวกไว้ในตัวแปร sum ผลบวก คือ 10 Please Enter N: 4 1 2 3 4

```
Sum from 1 to 4 = 10
```

**ตัวอย่างที่** 3 จงเขียนโปรแกรมเพื่อวนรับตัวเลขจากผู้ใช้จนกว่าผู้ใช้จะป้อนค่า ติดลบจึงจบการทำงาน

using System; class While4 { static void Main() { int N = 0; while (N >= 0) { // stop when N is negative Console.Write("Please input N: "); N = int.Parse(Console.ReadLine()); } Console.WriteLine("Bye Bye!!!"); } }

#### ผลลัพธ์ของโปรแกรม

Please input N: 3 Please input N: 2 Please input N: 3000 Please input N: 9999 Please input N: -50 Bye Bye!!!





## 9.1 คำสั่ง For loop

## 9.2 ตัวอย่างโปรแกรม

.1 🎽 คำสั่ง For loop

คำสั่ง For loop เป็นคำสั่งแบบทำงานวน ซ้ำ โดยจะมีการตรวจสอบเงื่อนไขก่อนทำตาม คำสั่งใน loopโดยที่ถ้าเงื่อนไขเป็นจริง จะ ทำงานในคำสั่งภายใน loop ไปเรื่อย ๆ จนกว่าเงื่อนไขเป็นเท็จจึงออกนอก loop

### 9.2 🎽 ตัวอย่างโปรแกรม

### **ตัวอย่างที่ 1** จงเขียนโปรแกรมเพื่อพิมพ์เลข 1 ถึง 20 บนหน้าจอภาพ

#### using System;

using System.Collections.Generic; using System.Linq; using System.Text; namespace ConsoleApplication16

#### class Program

#### {

static void Main(string[] args)

#### . ....

```
int i;
for (i = 1; i <= 20; i++)
    Console.WriteLine(i);
    Console.ReadKey();
```

ผลลัพธ์โปรแกรม 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

### **ตัวอย่างที่ 2** จงเขียนโปรแกรมเพื่อพิมพ์เลข 1 ถึง 5 โดยใช้คำสั่ง For loop พร้อมกับ หาผลบวกของเลข 1 ถึง 5 และแสดงผลออกบนหน้าจอภาพ



### <mark>ผลลัพธ์โปรแกรม</mark>





# 10.1 ชนิดข้อมูลแบบ Array 10.2 การอ้างถึงข้อมูลใน Array

10.3 ตัวอย่างโปรแกรม





### 10.1.1 การประกาศตัวแปรแบบอาเรย์

### ตัวแปรแบบอาเรย์จะต้องมีการประกาศล่วงหน้าก่อนการใช้งาน โดยมี รูปแบบดังต่อไปนี้

DataType[] ArrayName;

ในที่นี้ ArrayName คือ ชื่อตัวแปรแบบอาเรย์ และ DataType คือ ชนิดข้อมูลแต่ละ จำนวนที่ตัวแปรนี้เก็บค่าได้ สังเกตว่าการประกาศตัวแปรแบบอาเรย์มีความคล้ายคลึง กับการประกาศตัวแปรแบบปกติแต่ต่างกันเพียงเครื่องหมาย [] ที่ต้องใส่เพิ่มเข้าไป เท่านั้น **ตัวอย่างที่** 1 ส่วนของโปรแกรมต่อไปนี้เป็นการประกาศตัวแปรแบบอาเรย์สามตัว คือ score, grade และ Name เพื่อเก็บข้อมูลประเภทจำนวนเต็ม อักขระ และสตริง ตามลำดับ

int[] score; // score refers to an array of integers

char[] grade; // grade refers to an array of characters

string[] name; // Name refers to an array of strings

int s; // s refers to an integer

char c; // c refers to a character

string n; // n refers to a string

10.1.2 การสร้างอาเรย์

ตัวแปรแบบอาเรย์ที่ถูกประกาศขึ้นมาตามรูปแบบข้างต้นนั้นสามารถ นำไปใช้เพียงแค่อ้างถึงอาเรย์ชนิดนั้น ๆ เท่านั้น โดยคำสั่งที่เราจะใช้ใน กระบวนการนี้คือ คำสั่ง new ซึ่งมีรูปแบบการใช้งานดังนี้

new DataType[num\_elements]

ในที่นี้ DataType คือ ชนิดข้อมูล และ num\_elements คือ นิพจน์แบบจำนวน เต็มแสดงขนาดของอาเรย์ที่เราต้องการสร้าง การใช้งานคำสั่ง new ในรูปแบบข้างต้นนั้น อยู่ในรูปของนิพจน์ที่ให้ค่าอ้างอิงไปยังอาเรย์ที่เพิ่งถูกสร้างขึ้นมา ดังนั้น คำสั่ง new จึงมัก พบในรูป

ArrayName = new DataType[num\_elements];

โดยในที่นี้ ArrayName คือ ชื่อตัวแปรแบบอาเรย์ที่เราได้ประกาศเอาไว้ก่อนหน้านี้ และเช่นเดียวกับการประกาศและการให้ค่าเริ่มต้นกับตัวแปร

**ตัวอย่างที่ 2** การประกาศตัวแปรแบบอาเรย์และการสร้างอาเรย์ สร้างอาเรย์สำหรับ เก็บจำนวนเต็ม 5 ตัว โดยอ้างอิงผ่านตัวแปรแบบอาเรย์ชื่อ scores

int[ ] scores;

scores = new int[5];

ตามที่ได้กล่าวไว้แล้วข้างต้น เราสามารถรวมเอาการประกาศตัวแปรแบบอาเรย์และ การสร้างอาเรย์ไว้ในคำสั่งเดียวกันดังนี้

int[ ] scores = new int[5];

### สร้างอาเรย์สำหรับเก็บชื่อนักเรียน 10 คน อ้างอิงผ่านตัวแปรแบบอาเรย์ชื่อ names

string[] names;

names = new string[10];

หรือ

string[] names = new string[10];

**ตัวอย่างที่** 3 สร้างอาเรย์ชื่อ scores เพื่อเก็บจำนวนเต็ม 5 จำนวนโดยมีค่าเริ่มต้น เท่ากับ 10, 50, 10, 55 และ 60 ตามลำดับ

> int[] scores = new int[5] {10, 50, 10, 55, 60}; หรือ int[] scores = new int[] {10, 50, 10, 55, 60}; หรือ int[] scores = {10, 50, 10, 55, 60};

### 10.2) การอ้างถึงข้อมูลใน Array

เราสามารถเข้าถึงข้อมูลแต่ละจำนวนภายในอาเรย์ผ่านทางดัชนี (index) ที่ สอดคล้องกับตำแหน่งของข้อมูลนั้น ๆ โดยค่าดัชนีสำหรับข้อมูลตัวแรกสุดในอาเรย์ถูก กำหนดให้เป็นดัชนีหมายเลข 0 ถัดมาเป็นหมายเลข1 เรื่อยไป ในภาษาC# ข้อมูล ณ ตำแหน่ง idx ภายในอาเรย์ชื่อ ArrayName จะถูกอ้างถึงในรูปแบบ

ArrayName[idx]

**ตัวอย่างที่** 4 พิจารณาอาเรย์ scores ซึ่งประกอบด้วยจำนวนเต็ม 5 จำนวน กำหนดค่า 52 ให้กับข้อมูลตัวแรกของอาเรย์

scores[0] = 52;

น้ำข้อมูลในต่ำแหน่งท้ายสุดของอาเรย์มาแสดงผลบนหน้าจอ

Console.WriteLine(scores[4]);

กำหนดให้ค่าทุกค่าในอาเรย์มีค่าเท่ากับ 3

for (int i = 0; i < 5; i++)

scores[i] = 3;

คำนวณผลรวมของค่าทั้งหมดภายในอาเรย์

**int** sum = 0;

**for** (**int** i = 0; i < 5; i++)

sum = sum + scores[i];

น้ำค่าทั้งหมดในอาเรย์มาแสดงผลบนหน้าจอ

for (int i = 0; i < 5; i++)

Console.WriteLine("Student {0}: {1}", i+1, scores[i]);



**ตัวอย่างที่ 5** สร้างอาเรย์ชื่อ scores เพื่อเก็บจำนวนเต็ม 5 จำนวนโดยมี ค่าเริ่มต้นเท่ากับ 20, 40, 30,45 และ 55 ตามลำดับ และให้แสดงผลออกมาทาง หน้าจอภาพ

using System; using System.Collections.Generic; using System.Ling; using System.Text; namespace ConsoleApplication18 class Program static void Main(string[] args) int[] scores = new int[5] { 20, 40, 30, 45, 55 }; for (int i = 0; i < 5; i++)



}

Console.WriteLine("Student {0}: {1}", i + 1, scores[i]); Console.ReadKey();

### ผลลัพธ์โปรแกรม

Student 1 : 20 Student 2 : 40

Student 3 : 30

Student 4 : 45

Student 5 : 55

**ตัวอย่างที่ 6** สร้างอาเรย์ชื่อ scores เพื่อเก็บจำนวนเต็ม 5 จำนวนโดยมีค่าเริ่มต้น เท่ากับ 20, 40, 30,45 และ 55 ตามลำดับ และหาผลบวกของเลขจำนวนเต็ม 5 จำนวน โดยเก็บผลบวกไว้ในตัวแปร sum

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication19

class Program

static void Main(string[] args)

{

int[] scores = new int[5] { 20, 40, 30, 45, 55 };

```
for (int i = 0; i < 5; i++)
```

Console.WriteLine("Student {0}: {1}", i + 1, scores[i]);

int sum = 0; for (int i = 0; i < 5; i++) sum = sum + scores[i]; Console.WriteLine("sum {0}", sum); Console.ReadKey();

### ผลลัพธ์โปรแกรม

sum 190				
Student	5	:	55	
Student	4	:	45	
Student	3	:	30	
Student	2	:	40	
Student	1	:	20	

**ตัวอย่างที่ 7** สร้างอาเรย์สำหรับเก็บชื่อนักเรียน 5 คน อ้างอิงผ่านตัวแปรแบบ อาเรย์ชื่อ names และให้แสดงผลออกมาทางหน้าจอภาพ

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication20

class Program



```
{
  static void Main(string[] args)
  {
    string[] names;
    names = new string[5]{"Warapron","Weera","Khanidtha","Thakon","Pronchai"};
    for (int i = 0; i < 5; i++)
        Console.WriteLine("Student {0}: {1}", i + 1, names[i]);
        Console.ReadKey();
    }
}</pre>
```

### ผลลัพธ์โปรแกรม

Student	1	:	Warapron
Student	2	:	Weera
Student	3	:	Khanidtha
Student	4	:	Thakon
Student	5	:	Pronchai







11.1 ขั้นตอนการเขียนโปรแกรมด้วย Windows Forms Application 11.2 ตัวอย่างโปรแกรมพื้นที่สี่เหลี่ยมผืนผ้า 11.3 ตัวอย่างโปรแกรมทักทาย โดยใช้รูปแบบ MassageBox.show 11.4 ตัวอย่างโปรแกรมบวกเลข 11.5 ตัวอย่างโปรแกรมเปรียบเทียบอายุ โดยใช้เงื่อนไข if 11.6 ตัวอย่างโปรแกรมเลขคู่เลขคื่ โดยใช้เงื่อนไข if 11.7 ตัวอย่างโปรแกรมค่าคอมมิชชั้น โดยใช้คำสั่งเงื่อนไข if

### ขั้นตอนการเขียนโปรแกรมด้วย Windows Forms Application

### **ขั้นตอนที่ 1** เลือก New Project ดังหมายเลข 1

1.1

😎 Start Page - Microsoft Visual Studio						-	0	×
File Edit View Debug Team Data Tools Architecture Test Analyze Window Help								
Solution (1998) (199		- 2	) 🕾 🕠 🖄	🎾 🛃 🖻	]• =			
Start Page ×						Solution Explorer	•	ą×
						<b>B</b>		
Microsoft								
VISUAI STUCIO 2010 Ultimate								
Get Started Guidance and Resources Latest	News							
Welcome Windows Web Cloud Office Share	Point Data							
New Project								
Open Project What's New in Visual St	udio 2010							
Taxates Learn about the new feature	es included in this releas	ie.						
Recent Projects Visual Studio 2010 Overview	v vork 4							
ConsoleApplication21 What's New in Visual C#	WOIK 4				11			
况 ConsoleApplication20	Customize the Visual Studio Start Page					Properties	-	ąΧ
ConsoleApplication19								
Close page after project load						21 <u>21</u>		
Show page on startup								
Fron List					• a x			
O Errors 1 0 Warnings 1 0 Messages								
Description	File	Line	Column	Project				
					_			_
Ready								

**รูปที่ 11.1** เลือก New Project



....

-

-

-

**รูปที่ 11.2** เลือก Windows Forms Application ดังหมายเลข 2 กด ok



-

**รูปที่** 11.3 หน้าจอพร้อมเขียนโปรแกรม

หน้าต่าง Form Designer Form Designer เป็นหน้าต่าง ที่ใช้สำหรับการออกแบบหน้าต่างหรือหน้าตาของแอปพลิเค ชัน โดยการลากคอนโทรลต่างจาก Toolbox มาวางบน Form **ปุ่มกด (Button)** ใช้สำหรับคลิกเพื่อเลือกการทำงาน บางอย่าง

**ลาเบล** (Label) ใช้สำหรับแสดงข้อความ พิกเจอร์บ็อกซ์ (Picture box) ใช้สำหรับแสดงภาพ เท็กซ์บ็อกซ์ (Text box) ใช้สำหรับรับ/แสดงข้อมูลจาก คีย์บอร์ด


-

**รูปที่ 11.4** ตัวอย่างหน้าต่างโปรแกรมพื้นที่สี่เหลี่ยมผืนผ้า



-

**รูปที่ 11.5** การตั้งค่าโปรแกรมพื้นที่สี่เหลี่ยมผืนผ้า

😎 WindowsFormsApplication5 - Microsoft Visual Studio File Edit View Project Build Debug Team Data Format Tools Architecture Test Analyze Window Help	- 0 X
:[b] • 涵 • ❷ 및 ❷   ※ 函 逸   り • (P • 및 • 및   ▶ Debug • x86 •   ❷ •   ❷ •   嗯 音 局 数 次 動 國 □ • ; : ⊈   ▷ 수 릐   示 ↔ 프   금 別 弼 侍   ∞ 洪 밖 유   용 밝 왕 라   관 된   집 원   昌 國 ;	
Form1.cs [Design] × Form1.Designer.cs	<ul> <li>Solution Explorer</li> <li>Solution WindowsFormsApplication</li> <li>Solution "WindowsFormsApplication</li> <li>Properties</li> <li>References</li> <li>Form1.cs</li> <li>Form1.cs</li> <li>Program.cs</li> </ul>
คำนวณ ออกจากโปรแกรม 	Properties * # × Form1 System.Windows.Forms.F • RightToLeftL False Showlcon True Showlcon True Size 461, 382 Size 461, 382 Size 5ize 461, 382 Size 5ize 5ize 161, 382 5ize 5ize 161, 382 5ize 5ize 161, 382 1
Ready :	<u>_</u> 8 x 11

# **รูปที่ 11.6** หน้าต่างการตั้งค่าโปรแกรมพื้นที่สี่เหลี่ยมผืนผ้า

## การใส่โค้ดโปรแกรม

เราจะใช้ปุ่มคอนโทรลดังนี้

Label4 เปลี่ยน Properties Text เป็นคำว่า โปรแกรม พื้นที่สี่เหลี่ยมผืนผ้า

Label1 เปลี่ยน Properties Text เป็นป้อนความกว้าง Label2 เปลี่ยน Properties Text เป็นป้อนความยาว Label3 เปลี่ยน Properties Text เป็นพื้นที่เท่ากับ Button1 เปลี่ยน Properties Text เป็นคำนวณ Button2 เปลี่ยน Properties Text เป็นเคลียร์ค่า Button3 เปลี่ยน Properties Text เป็นออกจากโปรแกรม



#### อธิบายโปรแกรม

double width, length, area; เป็นการกำหนดตัวแปร width, length, area ให้เป็นแบบทศนิยม (double)

width = double.Parse(textBox1.Text); แปลงค่าใน textBox1 ให้เป็นทศนิยมแล้วนำมาเก็บไว้ ในตัวแปร width

length = double.Parse(textBox2.Text); แปลงค่าใน textBox2 ให้เป็นทศนิยมแล้วนำมาเก็บ ไว้ในตัวแปร length

area = width \* length; นำค่าที่เก็บไว้ในตัวแปร width มาคูณกับ ตัวแปร length แล้วนำ ผลลัพธ์ไปเก็บไว้ในตัวแปร area

textBox3.Text = area.ToString("##,###.00"); กำหนดรูปแบบการแสดงผลให้มีเครื่องหมาย คอมม่าและทศนิยม 2 ตำแหน่ง แล้วนำผลลัพธ์ไปแสดงผลไว้ที่ textBox3

หลังจากใส่โค้ดในปุ่มคำนวณแล้ว ให้รัน โปรแกรมเพื่อดูผลลัพธ์ โดยกดปุ่ม F5 <mark>ผลลัพธ์โปรแกรม</mark>



#### **รูปที่ 11.7** ตัวอย่างผลลัพธ์ของโปรแกรม



**โค้ดโปรแกรมปุ่มเคลียร์ค่า** textBox1.Clear(); {หมายถึง ให้เคลียร์ ค่าในคอนโทรล textBox2.Clear();

**รูปที่ 11.8** การเคลียร์ค่าโปรแกรมพื้นที่สี่เหลี่ยมผืนผ้า

**โค้ดโปรแกรมปุ่มออกจากโปรแกรม** ออกจากโปรแกรม (Button3) เมื่อคลิก ก็จะออกจากโปรแกรมทันที

Close();

textBox3.Clear();

#### 1.3 ตัวอย่างโปรแกรมทักทาย โดยใช้รูปแบบ MassageBox.show

### การใส่โค้ดโปรแกรม

เราจะใช้ปุ่มคอนโทรลดังนี้ Label1 เปลี่ยน Properties Text เป็นคำว่า โปรแกรมทักทาย Label2 เปลี่ยน Properties Text กรุณาป้อนชื่อ Label3 เปลี่ยน Properties Text กรุณาป้อนนามสกุล textBox1 ไว้รับข้อมูลชื่อ textBox2 ไว้รับข้อมูลนามสกุล Button1 เปลี่ยน Properties Text เป็นคำว่า ทักทาย

## ใ*ส่*โค้ดโปรแกรมไว้ในปุ่ม

ทักทาย

MessageBox.Show("สวัสดี " + textBox1.Text + " " + textBox2.Text);

เมื่อใส่โค้ดเสร็จแล้ว ให้กดปุ่ม F5 เพื่อดูผลลัพธ์โปรแกรม ในที่นี้ป้อนชื่อ วราภรณ์ นามสกุล อุ้มอังวะแล้วคลิกปุ่มทักทาย ก็จะมี MessageBox โชว์ขึ้นมาว่า "สวัสดี วราภรณ์ อุ้มอังวะ"

#### ผลลัพธ์โปรแกรม



**รูปที่ 11.9** ตัวอย่างหน้าต่างโปรแกรมทักทาย

#### ด้วอย่างโปรแกรมบวกเลข

11.4

## โค้ดโปรแกรมปุ่ม 🛛 บวก

#### int a, b, c;

private void button1\_Click(object sender, EventArgs e)

#### a = int.Parse(textBox1.Text);

b = int.Parse(textBox2.Text);

c = a + b;

textBox3.Text = c.ToString("##,###.00");

### ผลลัพธ์โปรแกรม

🚽 โปรแกรมบวกเลข				×
	Ĩ1	ไรแกรมบวกเลข		
	ป้อนจำนวนที่ 1	100		
	ป้อนจำนวนที่ 2	200		
	ป้อนจำนวนที่ 3	300.00		
		บวก		

**รูปที่ 11.10** โปรแกรมบวกเลข

ตัวอย่างโปรแกรมเปรียบเทียบอายุ โดยใช้เงื่อนไข if

โดยถ้าป้อนอายุ น้อยกว่า 60 จะขึ้นข้อความว่า "คุณเป็นเยาวชน" แต่ถ้าป้อน 60 ปีขึ้นไป จะขึ้นข้อความว่า "คุณเป็นเยาวชน"

โค้ดโปรแกรม ปุ่ม 🛛 ทดสอบ

11.5

#### age1 เป็นตัวเป็นที่ใช้เก็บอายุ

age1 = int.Parse(textBox1.Text); แปลงค่าใน textbox1 ให้เป็นเลขจำนวนเต็มแล้วนำมาเก็บไว้ใน ตัวแปร age1

int age1;

age1 = int.Parse(textBox1.Text);

if (age1 < 60)

{

label3.Text = "คุณเป็นเยาวชน";

else

£

label3.Text = "คุณเป็นผู้สูงอายุ";



## .6) ตัวอย่างโปรแกรมเลขคู่เลขคี่ โดยใช้เงื่อนไข if

โค้ดโปรแกรม ปุ่ม 🛛 ทดสอบ

int number;

number = int.Parse(textBox1.Text);

if (number % 2 == 0) textBox2.Text = "จำนวนคู่";

else

textBox2.Text = "จำนวนคี่";

## number เป็นตัวเป็นที่ใช้เก็บตัวเลข

number = int.Parse(textBox1.Text); แปลงค่าใน textbox1 ให้เป็นเลขจำนวนเต็มแล้วนำมาเก็บไว้ ในตัวแปร number





# ผลลัพธ์โปรแกรมที่ป้อนตัวเลขจำนวนคี่



**รูปที่** 11.14 ผลลัพธ์โปรแกรมที่ป้อนตัวเลขจำนวนคู่

ผลลัพธ์โปรแกรมที่ป้อนตัวเลขจำนวนคู่

 $\times$ 

**รูปที่** 11.13 ผลลัพธ์โปรแกรมที่ป้อนตัวเลขจำนวนคี่



ถ้ายอดขายรวม มากกว่า 6,000 บาท ขึ้นไป คิดคอมมิชชั่น 10%

โค้ดโปรแกรม ปุ่ม

คำนวณ

```
double sale1, sale2, sale3,total,commis;
private void button1_Click(object sender, EventArgs e)
{
    sale1 = double.Parse(textBox2.Text);
    sale2 = double.Parse(textBox3.Text);
    sale3 = double.Parse(textBox4.Text);
    total = sale1 + sale2 + sale3;
    if (total<=3000)
    {
        commis = total * 0.05;
```

```
textBox5.Text = total.ToString("##,###.00");
     textBox6.Text = commis.ToString("##,###.00");
   }
  else
     if (total <= 6000)
        commis = total * 0.07;
        textBox5.Text = total.ToString("##,###.00");
        textBox6.Text = commis.ToString("##,###.00");
      else
        commis = total * 0.10;
        textBox5.Text = total.ToString("##,###.00");
        textBox6.Text = commis.ToString("##,###.00");
private void button2_Click(object sender, EventArgs e)
  textBox1.Clear();
  textBox2.Clear();
  textBox3.Clear();
  textBox4.Clear();
  textBox5.Clear();
  textBox6.Clear();
```



	โปรแกรมค่	- ⊔ × ากอมมิสชั่น	
	ชื่อพนักงาน	วราภรณ์ อุ้มอังวะ	
	ยอดขายชนิดที่ 1	1000	
	ยอดขายชนิดที่ 3	3000	
	ยอดขายรวม	6,000.00	
	ค่าคอมมีสชั่น	420.00	
	ถ้านวณ เค	ลียด่า จบกา <del>ร</del> ทำงาน	
โค้ดโปรแกรม ปุ่	<b>ง จบการทำงาน</b> จบกา	ารทำงาน	
Close();			





12.1 คลาสและออบเจ็กต์
12.2 ออบเจ็กต์และไทป์
12.3 การใช้งานออบเจ็กต์
12.4 เอ็นแคปซูเลชัน
12.5 การสืบทอด
12.6 โพลีมอร์ฟิส

12.7 เมธอด

## 12.1 คลาสและออบเจ็กต์

คลาสและออบเจ็กต์ เป็นแนวคิดพื้นฐานของการเขียนโปรแกรมเชิงวัตถุ ซึ่งคลาสเป็น พิมพ์เขียว หรือต้นแบบที่ผู้ใช้กำหนดซึ่งวัตถุถูกสร้างขึ้น โดยพื้นฐานแล้วคลาสจะรวมฟิลด์ และเมธอด (ฟังก์ชันสมาชิกซึ่งกำหนดการกระทำ) ไว้ในหน่วยเดียว คลาสและออบเจ็กต์ เป็นสิ่งที่เกี่ยวข้องกันโดยตรง คลาสคือ โค้ด โปรแกรมที่เราเขียนขึ้นเพื่อทำหน้าที่เป็นพิมพ์ เขียวของออบเจ็กต์



โดยทั่วไปการประกาศคลาสสามารถรวมคอมโพเนนต์เหล่านี้ตามลำดับ

- 1. Modifiers : คลาสสามารถเป็นพับบลิกหรือไปรเวท ดีฟอลต์ของคลาส คือ พับบลิก
- 2. Keyword class : คำหลักที่ใช้ในการประกาศ
- Class Identifier : ตัวระบุ หรือ ชื่อของคลาสตัวอักษรเริ่มต้นซึ่งควรเป็นตัวพิมพ์ ใหญ่

4. Base class or Super class : ชื่อของพาเวนต์ของคลาส (ซูเปอร์คลาส) ถ้ามี นำหน้าด้วย : (โคลอน)

5. Interfaces : รายการของอินเทอร์เฟซที่คั่นด้วยเครื่องหมายจุลภาคที่นำไปใช้โดย คลาสหากมีนำหน้าด้วย: (โคลอน) คลาสสามารถใช้อินเทอร์เฟซได้มากกว่าหนึ่ง อินเทอร์เฟซ

6. Body : ร่างกายของคลาสนั้นล้อมรอบด้วย { } (เครื่องหมายปีกกา)

ในภาษาโปรแกรมที่ไม่ใช่ภาษาแบบ OOP เช่น ภาษา C จะมีไทป์ ต่าง ๆ ให้ใช้งาน เช่น int, float, double, char เป็นต้น ไทป์ก็คือชนิดข้อมูล เมื่อเรานิยามตัวแปรซึ่งมีชนิดข้อมูลใด ๆ ตามที่ ต้องการตัวแปลภาษา จะทราบทันที่ว่าต้องทำอย่างไรกับชนิดข้อมูลนั้น ๆ เนื่องจากว่าไทป์ เหล่านั้นเป็นไทป์ที่ได้นิยาม หรือ กำหนดไว้ล่วงหน้าแล้ว เราเรียกไทป์แบบนี้ว่าไทป์ภายใน ในภาษา C หากพบว่าไทป์ที่มีให้ไม่ตรงกับความต้องการของเรา เราสามารถนิยามไทป์ ขึ้นมาใช้งานเองได้ ซึ่งเราจะเรียกไทป์แบบนี้ว่า User Defined Data Type โดยใช้สตรักเจอร์ ซึ่ง การสร้างUDT ด้วยสตรักเจอร์ก็คือการนำไทป์ชนิดต่าง ๆ มาผสมกัน

คลาสก็เหมือนสตรักเจอร์ เรานิยามคลาสขึ้นมาใช้งานเพราะต้องการสร้าง UDT ใหม่นั่นเอง เมื่อเรานิยามไทป์ขึ้นมาใหม่ ไทป์แบบ UDT ก็จะถูกคอมไพเลอร์ปฏิบัติต่อไทป์ที่เราสร้างขึ้นแบบ เดี่ยวกับ Build inType .NET Framework Library นับว่าเป็นแหล่งรวบรวมไทป์มากมายหลายพัน ไทป์ ให้เราสามารถนำมาสร้างเป็นไทป์ชนิดใหม่ที่มีความ สามารถในการทำงานที่ซับซ้อนขึ้นได้ อย่างสะดวกและรวดเร็ว

### 12.3) การใช้งานออบเจ็กต์

ออบเจ็กต์เปรียบเสมือนกล่องดำ (Black Box) ซึ่งเวลาเรานำมาใช้งานเราไม่ จำเป็นต้องรู้ว่ามันมีกลไกลในการทำงานอย่างไร หรือ ไม่จำเป็นต้องรู้ว่าภายใน ออบเจ็กต์นั้นมีโค้ดอะไรภายใน เมื่อเราต้องการใช้งาน เราจะสั่งงานผ่านเมธอด (Method) ออบเจ็กต์จะผนวกขบวนการ (Method) และข้อมูล (Information) ไว้ ภายในตัวของมันเอง ออบเจ็กต์จะเก็บข้อมูลภายในไว้ในฟิลด์ (field) หรือ ตัวแปร ท้องถิ่นของคลาสนั่นเอง

โปรแกรมที่เรียกใช้ออบเจ็กต์ไม่จำเป็นและสามารถเข้าถึง หรือเปลี่ยนแปลง ข้อมูลในฟิลด์ได้โดยตรงแต่สามารถเข้าถึงได้โดยทางอ้อมผ่านส่วนเชื่อมต่อที่ เรียกว่าพร็อปเพอร์ตี้ (Property)



เอ็นแคปซูเลชัน หรือ คุณสมบัติการห่อหุ้ม การห่อหุ้ม คือ แนวคิดของออบเจ็กต์ที่ แยกโค้ดส่วนเชื่อมต่อ กับโค้ดส่วนประมวลผลออกจากกัน ข้อมูลและส่วนประมวลผล ทั้งหมดจะถูกซ่อนไว้เบื้องหลังอินเทอร์เฟส หากเราจัดให้มีการอินเทอร์เฟสที่ดีและรักษา รูปแบบการเชื่อมต่อไว้อย่างคงที่



## 12.4.1 รวมส่วนข้อมูลและส่วนประมวลผลไว้ด้วยกัน

ส่วนของข้อมูลคลาสเราจะเรียกส่วนนี้ว่า Data Field หรือ Variable Filed ทำหน้าที่เก็บข้อมูล ในส่วนของการประมวลผลซึ่งเป็นสมการคณิตศาสตร์อย่างง่าย ๆ ที่ทำหน้าที่แปลงหน่วยวัดอุณหภูมิจากองศาเซลเซียสไปเป็นองศาฟาเรนไฮต์การ รวมข้อมูลและส่วนประมวลผลไว้ด้วยกันแบบนี้คือลักษณะอย่างหนึ่งของหลักการ เอ็นแคปซูเลชัน

# 12.4.2 ใช้กลไลในการเชื่อมกับโค้ดภายนอกด้วยพร็อปเพอร์ตี้

หากเราสามารถกันไม่ให้โค้ดภายนอก สามารถเข้าถึงส่วนของข้อมูล ภายในออบเจ็กต์ได้ คลาสนั้น ๆ ก็มีคุณสมบัติของการเอ็นแคปซูเลชัน ทำให้ เราสามารถส่งค่าองศาเซลเซียสไปให้ออบเจ็กต์ได้โดยไม่มีการแตะต้องส่วน ของ Data Filed ของคลาสเลย

## (12.5) การสืบทอด

วิธีการอย่างง่าย ๆ ในการตรวจสอบดูว่าภาษาคอมพิวเตอร์ใดสนับสนุนแนวความคิดใน แบบ OOPหรือไม่นั้น ให้ตรวจสอบดูว่าภาษาคอมพิวเตอร์ภาษานั้นมีคุณสมบัติของการสืบ ทอด หรือ การอินเฮียริแตนซ์หรือไม่ เรียกย่อ ๆ ว่าอินเฮียริต หลักการสำคัญนี้มีไว้เพื่อให้เรา สามารถต่อยอดงานใหม่จากงานเดิมที่มีอยู่แล้วนั่นเอง



การแปรคุณสมบัติ คำว่าโพลี (Poly) แปลว่าหลาย และคำว่ามอร์ฟิส แปลว่า รูปร่าง ดังนั้นคำว่าโพลีมอร์ฟิสนี้จึงแปลว่าหลายรูปทรง หรือ เปลี่ยนสภาพได้ อะไรที่ เปลี่ยนรูปร่างได้คำตอบ คือ เมธอดที่อินเฮียริตมาจากเบสคลาส

โพลีมอร์ฟิส

12.6



**รูปที่ 12.5** Polmorphism

## ตัวอย่างแสดงการทำเมธอดโอเวอร์ไรท์อย่างง่าย ๆ มีดังนี้

#### Class CustomerData

}

protected virtual void CustomerPhone()

Cosole.WriteLine("02 123 456");

protected virtual void ClearAll()

Cosole.WriteLine("deleat all data");

### สร้างคลาสใหม่โดยการอินเฮียริตจากคลาส CustomerData

Class CustomerData2 : CustomerData
{
 protected override void CustomerPhone()
 {
 Base. CustomerPhone()
 Cosole.WriteLine("02 123 456 7");
 }
 protected new virtual void ClearAll()
 {
 Cosole.WriteLine("deleat all data");
 }
}



7) เมธอด

เมธอด (Method) เป็นองค์ประกอบอย่างหนึ่งของคลาส สำหรับกำหนด รูปแบบ วิธีการกระทำอย่างใดอย่างหนึ่ง

## 12.7.1 การสร้างเมธอด

จากตัวอย่างที่ผ่านมามีเมธอดที่เราได้รู้จัก คือ Main()

using System;

using System.Collections.Generic; using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

//...

.

}

โดยทั่วไปแล้วคลาสมักประกอบไปด้วยเมธอดมากกว่า 1 เมธอด สำหรับ กำหนดการทำงานเฉพาะอย่างนั่น คือ เราสามารถเพิ่มเมธอดให้กับคลาสได้ จะกี่ เมธอดก็ได้ตามความจำเป็น หากเราเพิ่มเมธอดลงไปในคลาสจะมีลักษณะดังนี้

> using System; using System.Collections.Generic; using System.Text;

namespace ConsoleApplication1

class Program

{

static void Main(string[] args)

//...

static (returnValueType, void) MethodName(parameter (s))

//...

จากเมธอดที่ได้เพิ่มเติมลงไป สามารถอธิบายได้ดังนี้ 1. returnValueType ใช้ในกรณีที่เมธอดนั้นจำเป็นต้องส่งผลลัพธ์ที่ประมวลผล ภายในเมธอดนั้นกลับไปยังส่วนที่เรียกใช้ เราจำเป็นต้องระบุชนิดข้อมูลเสมอ

> using System; using System.Collections.Generic; using System.Text; namespace ConsoleApplication1 class Program static void Main(string[] args) //... static double CalCircleArea(int r) //...

## 2. ส่งผลลัพธ์ด้วย return

.

using System; using System.Collections.Generic; using System.Text; namespace ConsoleApplication1 class Program static void Main(string[] args) //... static double CalCircleArea(int r) Double area = 3.14 \* r \* r; Return area;

## **3. การรับพารามิเตอร์** กรณีที่เมธอดมีหลายพารามิเตอร์จะเป็นดังนี้

using System; using System.Collections.Generic; using System.Text;

namespace ConsoleApplication1

class Program

static void Main(string[] args)

//...

static double CalXYZ(int x, double y, string z)

```
//...
```

}

l

# 12.7.2 การเรียกใช้เมธอด

การเรียกใช้เมธอดมี 2 แบบ คือ

 การเรียกใช้ void method หรือ เมธอดไม่คืนค่า กรณีนี้เพียงระบุชื่อเมธอด และพารามิเตอร์(ถ้ามี) เช่น

String msg = "Hello World;

Say(msg);

## การเรียกใช้เมธอดแบบคืนค่า จำเป็นต้องมีตัวแปรมารองรับค่าที่ส่งกลับ เช่น

Double myArea = CalCircleArea(10);

```
using System;
using System.Collections.Generic;
using System.Text;
namespace ConsoleApplication3
  class Program
     static void Main(string[] args)
        double radius; string s;
        Console.Write("Please enter radius : ");
        s = Console.ReadLine();
        radius = double.Parse(s);
        Console.WriteLine("Area = {0}, + Circumference = {1}"
           ,CalArea(radius),CalCircumference(radius));
        Console.Read();
     static double CalArea(double r)
        return 3.14 * r * r;
     static double CalCircumference(double r)
        return 2 * 3.141 * r;
```

Please enter radius : 100

Area = 13140, + Circumference = 628.2
### **้ฝึกปฏิบัติการสร้าง Class และเรียกใช้งาน Class C#** 1. สร้าง Class ในแบบ Windows App ดังนี้

### 1.1 สร้างโปรเจ็กต์ใหม่ในแบบ Windows App



**รูปที่ 12.6** การสร้างโปรเจ็กต์ใหม่ในแบบ Windows App



2

.....

🕅 WindowsFormsApplication1 - Microsoft Visual Studio (Administrator) $ abla = e x x$								
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP								
0								
Tradition			Add New Item - WindowsFormsAppli	cation1	? ×	- 1 - 1		
Casada	Teelleev Or	d lastellad	···	5		τ + P		
	Nindows Forms	- instancu	Sort by: Default	2	Search Installed Templates (Ctrl+E)	P - 35		
Ally	Pointer	▲ Visual C# Items	Class	Visual C# Items	Type: Visual C# Items	- م		
	BackgroundWorker	Code	247		An empty class definition	ion1' (1 project)		
20	BindingNavigator	General	••• Interface	Visual C# Items				
ű.	BindingSource	⊳ Web	_					
80	Button	Windows Forms	E Windows Form	Visual C# Items				
$\checkmark$	CheckBox	WPF		1041				
83	CheckedListBox	Reporting	User Control	Visual C# Items				
2	ColorDialog	SQL Server	Component Class	Visual C# Items				
₿	ComboBox	Workflow	Stg Component class	risder en rierris				
智	ContextMenuStrip	Graphics	User Control (WPF)	Visual C# Items				
ġ.	DataGridView	▷ Online				• 1 v		
1	DataSet		About Box	Visual C# Items		Propertier		
÷	DateTimePicker		49			ropenies		
27	DirectoryEntry		ADO.NET Entity Data Model	Visual C# Items				
٩	DirectorySearcher		Application Configuration File	Visual C# Iteres				
	DomainUpDown			visual C# items		IndowsFormsApplication Lcsproj		
፼	ErrorProvider		Application Manifest File	Visual C# Items		(Testelass (Windows) of this Applicatio		
6	EventLog							
69	FileSystemWatcher			10 10 AB				
	FlowLayoutPanel		Click here to go online and find templates.					
	FolderbrowserDialog	Name: Class1.cs						
87	GroupBox	Nome.						
	HelpProvider				Add Cancel			
	HScrollBar				-	7		
Server	Sever Employer Toolbox Error List Output							
	← 🥹 👑 🕅							

**รูปที่ 12.7** คลิก Add



Ŕ

**รูปที่ 12.8** Class1

2. สร้าง Class ในแบบ Console App ดังนี้

### 2.1 สร้างโปรเจ็กต์ใหม่ในแบบ Windows App



**รูปที่ 12.9** สร้างโปรเจ็กต์ใหม่ในแบบ Windows App

### 2.2 คลิก Add



. ...

**รูปที่ 12.10** คลิก Add

2.3 คลิก Add

.

-+ 5









1

13.2 เมธอดในการสร้างกราฟิก

# 13.1 ระบบพิกัดด้านกราฟิก

13.4 การวาดเส้นตรง

13.6 การวาดรูปสี่เหลี่ยม

13.3 ออบเจ็กต์พื้นฐานทางกราฟิก

13.5 การวาดเส้นโค้ง

13.8 การวาดวงกลมและวงรี

13.10 การวาดตัวอักษร

13.7 การวาดรูปหลายเหลี่ยม



#### 3.1 🔪 ระบบพิกัดด้านกราฟิก

ใน GDI+ จะมีระบบพิกัดสำหรับใช้งาน 3 ระบบ คือ World coordinate, Page coordinate และDevice coordinate

1. World coordinate ระบบหน่วยที่ใช้ในชีวิตประจำวัน เช่น หน่วยมิลลิเมตร หรือ หน่วยนิ้ว เป็นต้น

2. Page coordinate ระบบ coordinate บนหน้ากระดาษ

3. Device coordinate ระบบ coordinate บนจอภาพ หรือ เครื่องพิมพ์



### 13.2 เมธอดในการสร้างกราฟิก

พื้นฐานการสร้างออบเจ็กต์กราฟิกจะเรียกผ่านเมธอด CreateGraphics ของฟอร์ม หรือคอนโทรลตัวอื่น ๆซึ่งอยู่ภายใต้เนมสเปซ (Namespace) using System.Drawing; โดยมีรูปแบบการอ้างอิงของออบเจ็กต์ดังนี้

Graphics g = e.CreateGraphics();

#### ตัวอย่าง

private void button1\_Click(object sender, EventArgs e)

{

**Graphics** graph = CreateGraphics();



Color ใช้สำหรับกำหนดสีกำหนดผ่านออบเจ็กต์ Color เช่น Color.Green หรือ ใช้ผสม สี เช่นColor. FormArgb(Red, Green, Blue) หรือ Color.FormArgb(255 , 100, 50)

Pen ใช้สำหรับกำหนดลักษณะของเส้นที่จะวาด กำหนดผ่านออบเจ็กต์ Pen เช่น MyPen = NewPen (color, width)

Brush ใช้สำหรับระบายสี่ กำหนดผ่านออบเจ็กต์ Brush เช่น MyBrush = New SolidBrush(color)

Rectangle ใช้สำหรับวาดรูปทรง กำหนดผ่านออบเจ็กต์ Rectangle เช่น Rect = New Rectangle (x1,y1,width,height)

Point ใช้สำหรับกำหนดพิกัด หรือตำแหน่ง x , y กำหนดผ่านออบเจ็กต์ Point เช่น P = NewPoint(x,y)

### .3.4 การวาดเส้นตรง

### การวาดเส้นจะใช้เมธอด DrawLines โดยมีรูปแบบดังนี้

Graphics.DrawLines(penCurrent, Coordinates);

### ตัวอย่าง

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Drawing.Drawing2D;

```
.).
```

```
namespace WindowsFormsApplication1
```

public partial class Form1 : Form

```
public Form1()
```

```
{
InitializeComponent();
```

```
}
```

private void Form1\_Paint(object sender, PaintEventArgs e)

Pen pnBorder;

Color clr = Color.Red;

pnBorder = new Pen(clr);

Point p1 = new Point(10, 10);

Point p2 = new Point(300, 10);

e.Graphics.DrawLine(pnBorder, p1, p2);

#### 5

private void Form1\_Load\_1(object sender, EventArgs e)
{
 Paint += new PaintEventHandler(Form1\_Paint);





#### 13.5) การวาดเส้นโค้ง

### การวาดเส้นจะใช้เมธอด DrawCurve โดยมีรูปแบบดังนี้

Graphics.DrawCurve(Pen pen, Point[] points);

### ตัวอย่าง

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

```
using System.Drawing.Drawing2D;
```

namespace WindowsFormsApplication1

```
public partial class Form1 : Form
```

```
{
```

public Form1()

1

InitializeComponent();

```
}
```

private void Form1\_Paint(object sender, PaintEventArgs e)
{

Pen penCurrent = new Pen(Color.Blue);

Point[] pt = { new Point( 40, 42), new Point(188, 246),new Point(484, 192),

new Point(150, 240),new Point(404, 18);

e.Graphics.DrawCurve(penCurrent, pt);

}

private void Form1\_Load\_1(object sender, EventArgs e)
{
 Paint += new PaintEventHandler(Form1\_Paint);





# 13.6 การวาดรูปสี่เหลี่ยม

การวาดเส้นจะใช้เมธอด DrawRectangle โดยมีรูปแบบดังนี้

Graphics.DrawRectangle(penCurrent, Rect);

# ตัวอย่าง

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

```
using System.Drawing.Drawing2D;
```

namespace WindowsFormsApplication1

public partial class Form1 : Form

public Form1()

InitializeComponent();

```
}
```

....

private void Form1\_Paint(object sender, PaintEventArgs e)

Pen penCurrent = new Pen(Color.Red); Rectangle Rect = new Rectangle(20, 20, 248, 162);

e.Graphics.DrawRectangle(penCurrent, Rect);

private void Form1\_Load\_1(object sender, EventArgs e)
{
 Paint += new PaintEventHandler(Form1\_Paint);





### 13.7) การวาดรูปหลายเหลี่ยม

การวาดเส้นจะใช้เมธอด DrawPolygon โดยมีรูปแบบดังนี้

Graphics.DrawPolygon(Penpen, PointF[] points);

# ตัวอย่าง

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

```
using System.Windows.Forms;
using System.Drawing.Drawing2D;
```

namespace WindowsFormsApplication1

```
public partial class Form1 : Form
```

public Form1()

InitializeComponent();

private void Form1\_Paint(object sender, PaintEventArgs e)

{

Point[] Pt = { new Point(20, 50), new Point(180, 50), new Point(180, 20), new Point(230, 70), new Point(180, 120), new Point(180, 90), new Point(20, 90) };

Pen penCurrent = new Pen(Color.Red); e.Graphics.DrawPolygon(penCurrent, Pt);

private void Form1\_Load\_1(object sender, EventArgs e)
{
 Paint += new PaintEventHandler(Form1\_Paint);

.

WindowsFormsApp	ication5 (Running) - Microsoft Visual Studio (Administrator)		) P - 5 ×
FILE EDIT VIEW PROJ	CT BUILD DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP		Sign in 🔛
8 G - O 18 - 🖕 🖬 🖡	🎽 🎔 🔹 💎 🐂 📄 Continue 🗸 🕜 🔹 Debug 🔹 Any CPU 👘 🚽 🖉 🖉 🖉 No Application Insights Events 🛀 🕺 🔳 🗨 🏠 🏠	→ ら。 ら、 は 猛 Code Map   ¥	8 _ 6 6 3 2 4
Process: [1256] WindowsFo	msApplication' • 🗊 Lifecycle Events • Thread:	*	
		Ŧ	
Toolbox • I ×	Form1.cs @ +9 X Form1.cs [Design] @		IntelliTrace
Search Toolbox	Image: State	ander, PaintEventArgs e)	
▲ General There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.	<pre>     public formal()     {         InitializeComponent();      }         Ireference         private void Formal_Paint(object sender, PaintEventArgs e)         {             Point[] Pt = { new Point(20, 50), new Point(180, 50), new Point(180, 20),</pre>		To view IntelliTrace data, you must break execution of your application. II Break all More options: Open IntelliTrace settings Learn more about IntelliTrace
	110 % - 4	•	IntelliT Solutio Team E
Ready		Ln 25 Col 28	Ch 28 INS
🛋 🚯 🗮		▲ Pr1 10	из 18:04 из 4/7/2562

3

-



การวาดเส้นจะใช้เมธอด DrawEllipe โดยมีรูปแบบดังนี้

Graphics.DrawEllipse(Pen pen, Rectangle rect);

Graphics.DrawEllipse(Pen pen, int x, int y, int width, int height);

### ตัวอย่าง

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;



using System.Threading.Tasks;

using System.Windows.Forms;

using System.Drawing.Drawing2D;

namespace WindowsFormsApplication1

public partial class Form1 : Form

public Form1()

{

InitializeComponent();

}

private void Form1\_Paint(object sender, PaintEventArgs e)
{

Pen penCurrent = new Pen(Color.Red);

e.Graphics.DrawEllipse(penCurrent, new Rectangle(20, 20, 226, 226));

private void Form1\_Load\_1(object sender, EventArgs e)
{
 Paint += new PaintEventHandler(Form1\_Paint);





WindowsFormsApp FILE EDIT VIEW PROJ •••• III •• • III • Process: [5588] WindowsFor	lication5 (Running) - Microsoft Visual Studio (Administrator) CT BUILD DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP P ♥ ♥ ♥ ♥   ▷ Continue • ♡ • Debug • Any CPU •   ↓   ♪ ♥ No Application rmsApplication. • ⓒ ☞ Clebus • Thread:	n Insights Events 👻 🚽 🚺 🔳 🔕 🏻 1	♥         ♥         Quick Launch           10         →         <         C         Image: Code 1           1         =	n (Ctrl+Q)
Toolbox • 4 × Search Toolbox • • General There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.	<pre>Form1.cs # ** K Form1.cs [Design] #  WindowsFormsApplication5  InitializeComponent();  InitializeComponent();  Pen perivate void Form1_Paint(object sender, PaintEventArgs e) { Pen penCurrent = new Pen(Color.Red);  e.Graphics.DrawEllipse(penCurrent, new Rectangle(20, 20, 226, 226)); } ireference private void Form1_Load_1(object sender, EventArgs e) { Paint += new PaintEventHandler(Form1_Paint); } }</pre>	Paint(object     Form1_Paint(object     Form1	t sender, PaintEventArgs e)	<ul> <li>IntelliTrace</li> <li>IntelliTrace</li> <li>Image: Image: Image:</li></ul>
Pandu	91 % - 4  Fror List  V - 30 Errors 4.0 Warnings 0 Messages  Description	File Line	Search Error List Column Project ▲	IntelliT     Solutio     Team E
			C(178	I TI II

3

-

# (13.9) การไล่สี

### การวาดเส้นจะใช้เมธอด LinearGradientBrush โดยมีรูปแบบดังนี้

Graphics.LinearGradientBrush(Point point1,Point point2,Color color1,Color color2);

# ตัวอย่าง

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Drawing.Drawing2D;

namespace WindowsFormsApplication1

public partial class Form1 : Form

public Form1()

InitializeComponent();

}

private void Form1\_Paint(object sender, PaintEventArgs e)

LinearGradientBrush lgb = new LinearGradientBrush(new Point(20, 20),

new Point(450, 20),

Color.DarkGreen,

Color.LightBlue);

e.Graphics.FillRectangle(lgb, 20, 20, 430, 180);

private void Form1\_Load\_1(object sender, EventArgs e)

Paint += new PaintEventHandler(Form1\_Paint);

FILE EDIT VIEW PROJE	CT BUILD DEBUG TEAM TOOLS	TEST ARCHITECTURE ANALYZE WINDOW HEL	P ation Insights Events 👻	_ <b>II ■ 5</b> 10 → 4.	요 ৫ Z Code M	ap   <b>% _</b> i ta	Sign in 🔛	
Process: [2468] WindowsFor	msApplication! - E Lifecycle Events - Three	ead:	k Frame:	•	-			
Toolbox     # ×       Search Toolbox     P •       General     Image: Controls in this group.       Drag an item onto this text to add it to the toolbox.	Form1.cs # 4 × form1.cs [Design] # WindowsFormsApplication5 Enamespace WindowsFormsApplicatio f areferences public partial class Form1 : f ireference public Form1() f InitializeComponent() ireference private void Form1_Paint f LinearGradientBrush n g1%	<pre> * WindowsFormsApplication1.Form1 n1 Form ); (object sender, PaintEventArgs e) lgb = new LinearGradientBrush(new Point(20, 20), wew Point(450, 20), iolor.DarkGreen, olor.LightBlue); ngle(lgb, 20, 20, 430, 180);</pre>		Form1_Load_1(object sender, Ev	₹ rentArgs e)		IntelliTrace	
	Error List T • 30 Errors 1 0 Warnings	0 Messages	File	Search Err	or List Project 🔺	₽× ₽• IntelliT	Solutio Team E	
Ready					7 Col 1	Ch 1 III 👘 ሌ ENG	INS 18:59 4/7/2562	

🛛 การวาดตัวอักษร

การวาดเส้นจะใช้เมธอด DrawString โดยมีรูปแบบดังนี้

Graphics.DrawString(string s, Font font, Brush brush, float x, float y);

# ตัวอย่าง

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms; using System.Drawing.Drawing2D;

namespace WindowsFormsApplication1

public partial class Form1 : Form

public Form1()

InitializeComponent();

private void Form1\_Paint(object sender, PaintEventArgs e)

FontFamily fntFamily = new FontFamily("Times New Roman");
Font fntWrite = new Font(fntFamily, 24.00F, FontStyle.Regular);

e.Graphics.DrawString("Welcome to C# OOP", fntWrite, Brushes.Blue, 10.00F,

#### 8.00F);

}

private void Form1\_Load\_1(object sender, EventArgs e)
{
 Paint += new PaintEventHandler(Form1\_Paint);

totabas * # X   Search Looloox	WindowsFormsApp FILE EDIT VIEW PROJ © - ©   18 - 🖕 🖬 1 Process: [3696] WindowsFo	lication5 (Running) - Microsoft Visual Studio (Administrator) ICT BUILD DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW F P ♥ ♥ ♥ ♥ ↓ ▷ Continue • ♥ • Debug • Any CPU • ↓ ↓ # # ↓ # ♥ № App ImmsApplication! • ⓒ Lifecycle Events • Thread:	FELP Dication Insights Events ▾ Ț 🕌 🚺 ■ 🔕   ちン   ⇒ ら。 ら tack Frame:	Quick Launch (Ctrl+Q) Sign in Ct Za Code Map   % = the file   3 /3   ■ ; Ct Za Code Map   % = the file   3 /3   = the file   = the file   = the file
Error List	Toolbox     P       Search Toolbox     P       General     There are no usable controls in this group.       Drag an item onto this text to add it to the toolbox.	<pre>Form1.cs @ # X Form1.cs [Design] #  @ WindowsFormsApplication5</pre>	- Q <sub>a</sub> Form1_Paint(object sender, PaintEv  Form1 -   K  Welcome to C# OOP  10.00F, 8.00F);	IntelliTrace 4 X entArgs e)  Image: IntelliTrace data, you must break execution of your application.  Image: Imag
	eady	Error List <b>Y</b> • O Strors <b>Description</b>	Search Error L File Line Column	r # × ist ♪ r Project ▲ IntelliT Solutio Team E Col 53 Ch 53 INS 2000 ↔

3

â