

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญาโท



ชื่อหัวข้อ โปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0
Computer Assisted Instruction in Visual C++ 6.0

ชื่อนักศึกษา 1. นายชัยสิทธิ์ แสงไทยเจริญตาก รหัสประจำตัว 41031407
2. นางสาวอังคณา เสมาทอง รหัสประจำตัว 41031429

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชา อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ที่ปรึกษา อาจารย์อมรชัย ชัยชนะ

อาจารย์ที่ปรึกษาร่วม อาจารย์อำพล ทองระอา

คณะกรรมการสอบปริญญาโท		ลายมือชื่อ
1. อาจารย์อมรชัย	ชัยชนะ	
2. อาจารย์อำพล	ทองระอา	
3. อาจารย์ไพบุลย์	พวงวงศ์ตระกูล	
4. อาจารย์สุระชัย	พิมพ์สาตี	

วัน/เดือน/ปีที่สอบ วันพฤหัสบดีที่ 11 พฤษภาคม พ.ศ. 2543 เวลา 17.00 น.

สถานที่สอบ ห้อง ค.301 คณะครุศาสตร์อุตสาหกรรม สจล.

ในท.
๕๖๕๒
๕๖๕๒

ภาควิชารับรองแล้ว

ลงนาม.....

(ผศ.วิสุทธิ อธิพรธรรม)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม (ภาควิชาครุศาสตร์อุตสาหกรรม สจล.)
วันที่ 10 เดือน ๕ - พ.ศ. ๒๕๔๓



เลขหมึก.....
เลขทะเบียน..... 37186
วัน, เดือน, ปี..... 5 ก.ย. 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

โปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0

COMPUTER ASSISTED INSTRUCTION IN VISUAL C++ 6.0



ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์

ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง โปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0
Computer Assisted Instruction In Visual C++ 6.0

วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาการเขียนโปรแกรม Visual C++ 6.0
2. เพื่อออกแบบโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0
3. เพื่อสร้างโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0
4. เพื่อทดสอบโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0
5. เพื่อนำโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0 ไปใช้ในการเรียนการสอน

ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถใช้งานโปรแกรม Visual C++ 6.0 ได้
2. สามารถออกแบบโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0 ได้
3. สามารถสร้างโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0 ได้
4. สามารถสรุปผลการใช้งานโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0
5. สามารถนำโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0 ไปประยุกต์ใช้ในการเรียนและการสอนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	โปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0
นักศึกษา	นายชัยสิทธิ แสงไทยเจริญลาภ นางสาวอังคณา เสมาทอง
อาจารย์ที่ปรึกษา	อาจารย์อมรชัย ชัยชนะ
อาจารย์ที่ปรึกษาร่วม	อาจารย์อำพล ทองระอา
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชา	อิเล็กทรอนิกส์และคอมพิวเตอร์
ปีการศึกษา	2542

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ นำเสนอโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0 สร้างจากโปรแกรม Visual C++ 6.0 ซึ่งภายในโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0 นี้ จะประกอบไปด้วย การติดตั้งโปรแกรม, ความรู้พื้นฐานเกี่ยวกับภาษา C++, อาร์เรย์, พอยเตอร์, การใช้แบบฟอร์มและเครื่องมือใน โปรแกรม Visual C++ 6.0 และการติดต่อกับไฟล์ข้อความ โดยโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0 นี้ จะทำขึ้นเพื่อให้ผู้เรียน ได้เรียนรู้เนื้อหาด้วยตนเองตามความสามารถ ความสนใจ และความพร้อมของผู้เรียนเอง ซึ่งเป็นจุดประสงค์หลักที่ต้องการจะนำไปใช้แทนการสอนในระบบทั่วไป

II

Thesis Title	Computer Assisted Instruction In Visual C++ 6.0
Students	Mr.Chaisite Sangthaijaroenlap Ms.Aungkana Saymathong
Advisor	Mr.Amornchai Chaichana
Co-Advisor	Mr.Amphon Thongra-ar
Education Level	Bachelor of Science in Industrial Education
Program in	Electronics and Computer
Academic Year	1999

ABSTRACT

This thesis presents the Computer Assisted Instruction (CAI) in Visual C++ 6.0 by Visual C++ 6.0 with in the CAI Visual C++ 6.0 consist of Installation, Introduction to Visual C++ 6.0 and Text File. This CAI produces for learners study by themself to follow capable, interest and preparedness of them. It is the main object to want to instead of general teaching system.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริยญาานิพนธ์ฉบับนี้ สำเร็จลุล่วงลงได้ด้วยดี เนื่องมาจากความอนุเคราะห์ของอาจารย์ที่ปรึกษาปริยญาานิพนธ์ และอาจารย์ที่ปรึกษาปริยญาานิพนธ์ร่วม อาจารย์โกศล ตราชู ตลอดจนอาจารย์ประจำภาควิชาครุศาสตร์วิศวกรรมทุกท่าน ที่ได้กรุณาให้คำปรึกษาข้อเสนอแนะ ให้ความอนุเคราะห์ยืมหนังสือและโปรแกรมต่าง ๆ และเพื่อน ๆ ทุกคนที่ให้ความช่วยเหลือ และกำลังใจ ในการจัดทำโครงการนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IV

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	XI
สารบัญรูป	XII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 ชัดความสามารถของโครงการ	2
1.3 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎีและหลักการ	4
2.1 ความหมายของ CAI	4
2.2 คุณสมบัติของ CAI	4
2.3 การออกแบบบทเรียน CAI	5
2.3.1 บทเรียน CAI แบบศึกษาเนื้อหาใหม่	5
2.3.2 บทเรียน CAI แบบฝึกทบทวน	7
2.3.3 บทเรียน CAI แบบสร้างสถานการณ์จำลอง	7
2.3.4 บทเรียน CAI แบบเกมส์สอน	8
2.3.5 บทเรียน CAI แบบใช้ทดสอบ	8
2.4 การออกแบบโปรแกรมช่วยสอนด้วยคอมพิวเตอร์	8
2.4.1 ไร่้ความสนใจ	9
2.4.2 บอกวัตถุประสงค์	9
2.4.3 ทบทวนความรู้เดิม	10
2.4.4 การเสนอเนื้อหา	11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.4.5 ชี้นำทางการเรียนรู้	11
2.4.6 กระตุ้นการตอบสนอง	11
2.4.7 ให้ข้อมูลย้อนกลับ	12
2.4.8 ทดสอบความรู้หลังบทเรียน	12
2.4.9 การจำและแนะนำการใช้	12
2.5 ประโยชน์ของโปรแกรมช่วยสอนสำหรับผู้เรียน	12
2.6 ประโยชน์ของโปรแกรมช่วยสอนสำหรับผู้สอน	13
2.7 การพัฒนาของ CAI, Multimedia และ ITS จะเกี่ยวข้องกับระบบการศึกษา	13
2.7.1 ข้อดีของ CAI มากกว่าแบบสอนธรรมดา	13
2.7.2 ข้อจำกัด	13
2.7.3 การพัฒนาของ CAI	13
2.7.4 ตามแบบ IMS การใช้คอมพิวเตอร์	14
2.7.5 ภาพเคลื่อนไหว	14
2.8 แนวความคิดพื้นฐานของ CAI	14
2.8.1 การทำให้คอมพิวเตอร์ทำได้เท่าเทียมกับแบบที่ครูสอน	14
2.8.2 ประสิทธิภาพพิเศษ	14
2.8.3 โครงสร้าง	14
2.8.4 การพัฒนาให้ CAI ดึงดูดความสนใจได้	14
2.8.5 ผลกระทบของ CAI ในการสร้างบทเรียน	15
2.8.6 ข้อจำกัดของ CAI ในการสร้างบทเรียน	15
2.8.7 รูปแบบของการเรียนด้วยคอมพิวเตอร์	15
2.8.8 โครงสร้างเกี่ยวกับการสอน	16
2.8.9 การใช้	16
2.8.10 แบบฝึกหัด	16
2.9 ทฤษฎีเกี่ยวกับโปรแกรม Visual C++ 6.0	16
2.9.1 Visual Studio 98	17
2.9.2 เครื่องมือที่ใช้ในการพัฒนาโปรแกรมของ Visual Stud 98	18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง .	หน้า
2.9.3 ประวัติของโปรแกรม Visual C++ 6.0 และ MFC	18
2.9.4 โปรแกรม Visual C++ 6.0 Edition	20
2.10 การใช้งาน โปรแกรม Visual C++ 6.0	21
2.10.1 การเริ่มเข้าโปรแกรม Visual C++ 6.0	21
2.10.2 การออกจากโปรแกรม Visual C++ 6.0	23
2.10.3 โปรเจกต์เวิร์กสเปซ	23
2.10.4 การจัดการ โปรเจกต์เวิร์กสเปซ	24
2.10.5 รายละเอียดของแท็บต่างๆ ของโปรเจกต์เวิร์กสเปซ	26
2.10.6 คอมไพล์และลิงก์เกอร์	29
2.10.7 ทูลบาร์และเมนูบาร์ของโปรแกรม Visual C++ 6.0	30
2.10.8 รายละเอียด และการใช้งาน Help	33
2.10.9 การสร้างไคอะล็อก	34
2.10.10 การสร้าง Hotkey	36
2.10.11 App Wizard	37
2.10.12 การสร้างโปรแกรมแบบ Dialog-Based	39
2.10.13 การใช้งานไคอะล็อกอิดิตเตอร์	46
2.10.14 การวางคอนโทรลในไคอะล็อก	47
2.10.15 การกำหนดค่าให้กับคอนโทรล	47
2.10.16 การทดสอบไคอะล็อก	48
2.10.17 การโฟกัสของคอนโทรล	49
2.10.18 การแสดงข้อความในไคอะล็อก	49
2.10.19 การสร้างรูปภาพในไคอะล็อก	50
2.10.20 การแสดงรูปภาพที่เป็นไอคอน	51
2.10.21 การแสดงรูปภาพที่เป็นไฟล์รูปภาพ	55
2.10.22 การ Export รูปภาพ	57
2.10.23 การสร้างปุ่มกดในไคอะล็อก	58
2.10.24 Class Wizard	58

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.10.25 การใช้งานแท็กซ์บ็อกซ์บรรทัดเดียว	61
2.10.26 การจัดการข้อความในแท็กซ์บ็อกซ์	61
2.10.27 ทำความเข้าใจกับโค้ดโปรแกรม	65
2.10.28 การเปลี่ยนลำดับโฟกัส	66
2.10.29 การตรวจสอบข้อความในแท็กซ์บ็อกซ์	67
2.10.30 คลาสที่ใช้ในการจัดการกับแท็กซ์บ็อกซ์	68
2.10.31 การสร้างออบเจกต์คอนโทรล	69
2.10.32 การเลือกข้อความภายในแท็กซ์บ็อกซ์	70
2.10.33 การลบข้อความภายในแท็กซ์บ็อกซ์	72
2.10.34 การใช้งานฟังก์ชัน Copy, Paste และ Undo	73
2.10.35 การแทนที่ข้อความฟังก์ชัน ReplaceSel	75
2.10.36 การจำกัดจำนวนตัวอักษร	77
2.10.37 การกำหนดข้อความเริ่มต้นให้กับแท็กซ์บ็อกซ์	78
2.10.38 การกำหนดแท็กซ์บ็อกซ์ให้อยู่ในสถานะ ReadOnly	80
2.10.39 การสร้างแท็กซ์บ็อกซ์แบบ Password	83
2.10.40 การใช้งานแท็กซ์บ็อกซ์แบบหลายบรรทัด	85
2.10.41 ตัวเลือกในการสกออร์และการตัดคำ	87
2.10.42 ตัวเลือกอื่นๆ ของแท็กซ์บ็อกซ์	88
2.10.43 แมสเสจของแท็กซ์บ็อกซ์	89
2.10.44 การใช้ DDX จัดการข้อมูลในแท็กซ์บ็อกซ์	89
2.10.45 Check และ Radio box	95
2.10.46 การสร้างและการใช้งานเช็คบ็อกซ์	95
2.10.47 การสร้าง และการใช้งานเรดิโอบ็อกซ์	98
2.10.48 การสร้าง และควบคุมลิสต์บ็อกซ์	103
2.10.49 คลาสที่ใช้จัดการกับลิสต์บ็อกซ์	108
2.10.50 การจัดการแมสเสจของลิสต์บ็อกซ์	109
2.10.51 การจัดการลิสต์บ็อกซ์ด้วย DXX	112

สารบัญ (ต่อ)

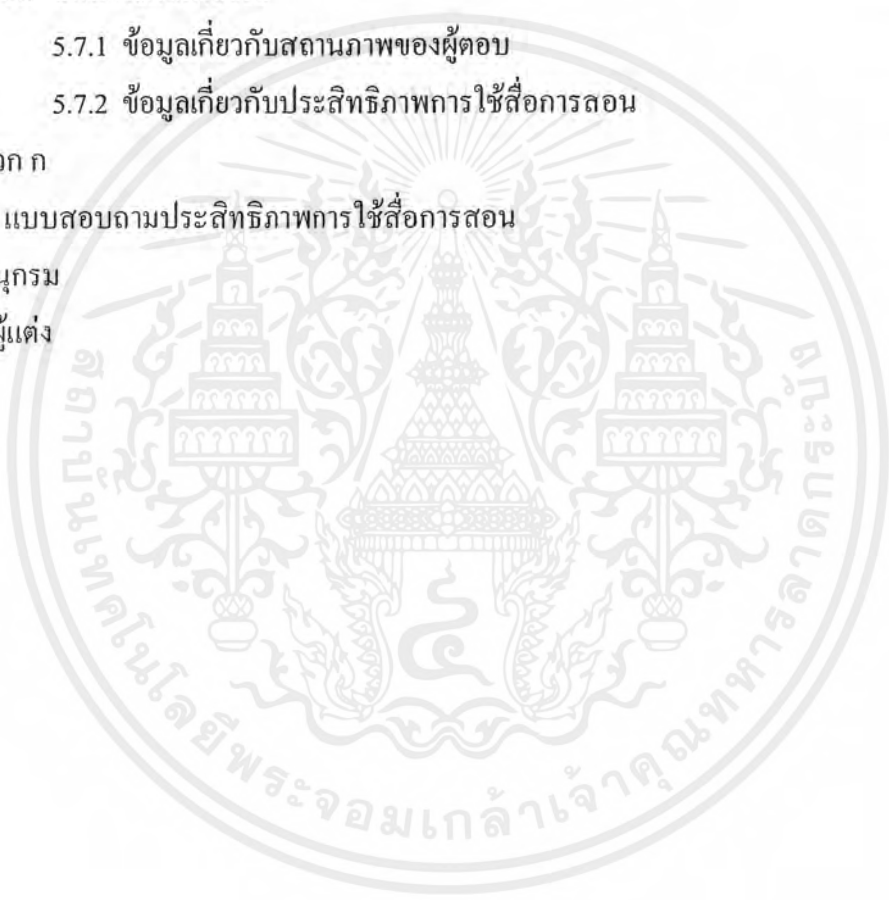
เรื่อง	หน้า
2.10.52 การสร้างและการใช้งานคอมโบบ็อกซ์	113
2.10.53 แมสเสจของคอมโบบ็อกซ์	115
2.10.54 ไดอะล็อก	115
2.10.55 การจัดการแมสเสจของไดอะล็อก	119
2.10.56 การติดตั้งเมนูบาร์ และเมนูป๊อปอัพให้กับไดอะล็อก	120
2.10.57 การใช้งาน Class Wizard แม็บแมสเสจให้กับเมนู	124
2.10.58 การสร้างเมนูป๊อปอัพโดยการคลิกขวา	127
2.10.59 การสร้างไดอะล็อกใหม่ให้กับโปรแกรม	128
2.10.60 การสร้าง และการใช้งานสปีนคอนโทรล	129
2.10.61 การนำค่าในสปีนคอนโทรลไปใช้	132
2.10.62 การสร้าง และการใช้งานสไลเดอร์คอนโทรล	134
2.10.63 แมสเสจบ็อกซ์	137
2.10.64 การจัดการเท็กซ์ไฟล์โดยใช้คลาสจาก MFC	139
2.10.65 การอ่านข้อมูลจากเท็กซ์ไฟล์	140
2.10.66 การบันทึกข้อมูลลงในไฟล์	142
บทที่ 3 การออกแบบ การสร้าง และการทำงาน	143
3.1 เครื่องมือที่ใช้ในการออกแบบ โปรแกรมช่วยสอน	143
3.1.1 ทางด้านฮาร์ดแวร์	143
3.1.2 ทางด้านซอฟต์แวร์	143
3.2 ขั้นตอนการออกแบบ โปรแกรม	144
3.3 โครงสร้างของโปรแกรม	145
3.4 โครงสร้างของโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0	146
3.5 บทเรียน	147
3.6 แบบทดสอบ	148
3.7 แบบทดสอบรวม	149
3.8 การสร้างโปรแกรม	150
3.8.1 การสร้างเมนูหลัก	151

สารบัญ (ต่อ)

เรื่อง	หน้า
3.8.2 การสร้างแบบทดสอบ	156
3.8.3 การสร้างแบบทดสอบรวม	157
3.8.4 การกำหนดโปรแกรมสามารถแสดงภาษาไทยได้	159
3.8.5 การสร้างแอปพลิเคชันภาษาไทย	162
3.8.6 การแม่ปแม่สเสงให้โครงการ	163
บทที่ 4 การทดลอง และผลการทดลอง	164
4.1 ความต้องการของโปรแกรม	164
4.2 การเข้าสู่โปรแกรม	164
4.3 การลงทะเบียนเพื่อการเข้าสู่บทเรียน	165
4.3.1 ผู้ใช้ยังไม่เคยลงทะเบียนมาก่อน	165
4.3.2 ผู้ใช้ผ่านการลงทะเบียนมาแล้ว	167
4.4 เมนูหลัก	168
4.5 การเลือกบทเรียน	168
4.6 การเข้าสู่บทเรียน	178
4.7 การคัดลอกตัวอย่างโปรแกรม	178
4.8 แบบทดสอบท้ายบทเรียน	178
4.9 แบบทดสอบรวม	181
4.10 ผลการตอบแบบสอบถาม	184
4.10.1 ตอนที่ 1 ข้อมูลเกี่ยวกับสถานภาพของผู้ตอบ	184
4.10.2 ตอนที่ 2 ข้อมูลเกี่ยวกับประสิทธิภาพของโปรแกรมช่วยสอน	187
บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา	196
5.1 สรุปผลของโครงการ	196
5.2 ปัญหาที่พบในการทำงาน	197
5.3 การแก้ปัญหา	197
5.4 แนวทางในการพัฒนาสรุปผลของโครงการ	197
5.5 รายงานการวิจัย	198
5.6 ผลการวิจัยแบบสอบถาม	198

สารบัญ (ต่อ)

เรื่อง	หน้า
5.6.1 ข้อมูลเกี่ยวกับสถานภาพของผู้ตอบ	198
5.6.2 ข้อมูลเกี่ยวกับประสิทธิภาพการใช้สื่อการสอน	198
5.7 อภิปรายผลการวิจัย	200
5.7.1 ข้อมูลเกี่ยวกับสถานภาพของผู้ตอบ	200
5.7.2 ข้อมูลเกี่ยวกับประสิทธิภาพการใช้สื่อการสอน	200
ภาคผนวก ก	201
แบบสอบถามประสิทธิภาพการใช้สื่อการสอน	202
บรรณานุกรม	206
ประวัติผู้แต่ง	207



สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 ประวัติการพัฒนาของ MFC และ โปรแกรม Visual C++	19
ตารางที่ 4.1 สถานภาพของผู้ตอบ	184
ตารางที่ 4.2 เพศของผู้ตอบ	184
ตารางที่ 4.3 สถานศึกษาของผู้ตอบ	185
ตารางที่ 4.4 แผนกวิชาของผู้ตอบ	186
ตารางที่ 4.5 ระดับการศึกษาของผู้ตอบ	186
ตารางที่ 4.6 ประสบการณ์ในการใช้สื่อการสอนของผู้ตอบ	187
ตารางที่ 4.7 ประสิทธิภาพการใช้สื่อการสอนด้านทักษะความรู้	187
ตารางที่ 4.8 ประสิทธิภาพการใช้สื่อการสอนด้านการใช้งาน	189
ตารางที่ 4.9 ประสิทธิภาพการใช้สื่อการสอนด้านประสิทธิภาพอื่น ๆ	191
ตารางที่ 4.10 ประสิทธิภาพการใช้สื่อการสอนการหาค่าเฉลี่ยด้านทักษะความรู้	192
ตารางที่ 4.11 ประสิทธิภาพการใช้สื่อการสอนการหาค่าเฉลี่ยด้านการใช้งาน	193
ตารางที่ 4.12 ประสิทธิภาพการใช้สื่อการสอนการหาค่าเฉลี่ยด้านประสิทธิภาพอื่น ๆ	195

สารบัญรูป

รูป	หน้า
รูปที่ 2.1 การเข้าสู่โปรแกรม Visual C++ 6.0	22
รูปที่ 2.2 โปรแกรม Visual C++ 6.0	23
รูปที่ 2.3 การสร้างโปรเจกต์เวิร์กสเปซ	24
รูปที่ 2.4 โปรเจกต์ใหม่	26
รูปที่ 2.5 Class View	27
รูปที่ 2.6 Resource View	28
รูปที่ 2.7 File View	28
รูปที่ 2.8 Toolbar Standard	30
รูปที่ 2.9 Toolbar Build	30
รูปที่ 2.10 Toolbar Build MiniBar	31
รูปที่ 2.11 Toolbar Resource	31
รูปที่ 2.12 Toolbar Edit	31
รูปที่ 2.13 Toolbar Debug	31
รูปที่ 2.14 Toolbar DataBase	32
รูปที่ 2.15 หน้าต่างของ MSDN Library	34
รูปที่ 2.16 ทูลบาร์สำหรับสร้างไดอะล็อก	35
รูปที่ 2.17 การแก้ไขค่า ID และ Caption	36
รูปที่ 2.18 การเพิ่ม ฮ็อตคีย์ ให้ไดอะล็อก	37
รูปที่ 2.19 MFC AppWizard	38
รูปที่ 2.20 การสร้างโปรเจกต์เวิร์กสเปซเลือกเมนู	39
รูปที่ 2.21 การเลือกรูปแบบโปรแกรม	40
รูปที่ 2.22 การกำหนดรายละเอียดให้กับโปรแกรม	42
รูปที่ 2.23 การกำหนดตัวเลือกของโปรเจกต์	43
รูปที่ 2.24 การสิ้นสุดการสร้างโปรเจกต์	45
รูปที่ 2.25 รายละเอียดของโปรเจกต์ที่สร้างขึ้น	46
รูปที่ 2.26 การวางคอนโทรลลงในไดอะล็อก	47
รูปที่ 2.27 การกำหนดค่าคงที่ให้กับคอนโทรล	48

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 2.28 การกำหนดลำดับโฟกัสคอนโทรล	49
รูปที่ 2.29 การแสดงข้อความลงในไดอะล็อก	50
รูปที่ 2.30 การสร้างไอคอนริชอร์สใหม่	52
รูปที่ 2.31 การกำหนดค่า ID ให้กับไอคอนริชอร์ส	53
รูปที่ 2.32 การสร้างสแตติกคอนโทรลเพื่อแสดงรูปไอคอน	54
รูปที่ 2.33 การกำหนดรูปภาพให้กับคอนโทรล	54
รูปที่ 2.34 ผลการรันโปรแกรมรูปไอคอน	55
รูปที่ 2.35 การ Import รูปภาพ	56
รูปที่ 2.36 การกำหนดรูปภาพให้กับไดอะล็อก	57
รูปที่ 2.37 การปรับแต่งปุ่มกด	58
รูปที่ 2.38 ClassWizard	59
รูปที่ 2.39 การเม็ปแมสเสจปุ่มกด	60
รูปที่ 2.40 การสร้าง Text Box บรรทัดเดียว	61
รูปที่ 2.41 การเม็ปแมสเสจให้ปุ่มกด	62
รูปที่ 2.42 ฟังก์ชันที่ ClassWizard สร้างขึ้น	63
รูปที่ 2.43 การคอมเมนต์คำสั่ง	63
รูปที่ 2.44 การเพิ่มโค้ดโปรแกรม	63
รูปที่ 2.45 ผลการรันโปรแกรม Text Box บรรทัดเดียว	64
รูปที่ 2.46 การใช้คำสั่ง MessageBox()	65
รูปที่ 2.47 การประกาศตัวแปร	65
รูปที่ 2.48 ฟังก์ชัน GetDlgItem()	66
รูปที่ 2.49 ฟังก์ชัน MessageBox()	66
รูปที่ 2.50 การทำงานของโปรแกรม Text Box บรรทัดเดียว	66
รูปที่ 2.51 โปรแกรมตรวจสอบข้อความในเท็กซ์บ็อกซ์	68
รูปที่ 2.52 โปรแกรมการตรวจสอบข้อความในเท็กซ์บ็อกซ์อีกแบบหนึ่ง	68
รูปที่ 2.53 การสร้างออบเจกต์คอนโทรล	69
รูปที่ 2.54 การใช้ฟังก์ชัน SetFocus()	70

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 2.55 รูปแบบของคำสั่ง SetSel()	70
รูปที่ 2.56 ตัวอย่างการใช้ฟังก์ชัน SetSel()	71
รูปที่ 2.57 ผลการรันการเลือกข้อความ	71
รูปที่ 2.58 การเลือกข้อความทั้งหมดในเท็กซ์บ็อกซ์	72
รูปที่ 2.59 ผลการรันการเลือกข้อความทั้งหมด	72
รูปที่ 2.60 การลบข้อความในเท็กซ์บ็อกซ์โดยใช้ฟังก์ชัน Clear()	73
รูปที่ 2.61 การลบข้อความในเท็กซ์บ็อกซ์โดยใช้ฟังก์ชัน Cut()	73
รูปที่ 2.62 การใช้ฟังก์ชัน Copy()	73
รูปที่ 2.63 การใช้ฟังก์ชัน Paste()	74
รูปที่ 2.64 เมนูป๊อปอัพ	75
รูปที่ 2.65 รูปแบบการใช้งานฟังก์ชัน ReplaceSel()	75
รูปที่ 2.66 ตัวอย่างการใช้ฟังก์ชัน ReplaceSel()	76
รูปที่ 2.67 ผลการรันการใช้ฟังก์ชัน ReplaceSel()	76
รูปที่ 2.68 การกำหนดให้ข้อความที่ถูกแทนที่สามารถ Undo ได้	77
รูปที่ 2.69 การจำกัดตัวอักษร	78
รูปที่ 2.70 การกำหนดข้อความเริ่มต้นให้กับเท็กซ์บ็อกซ์	78
รูปที่ 2.71 ผลการรันการกำหนดค่าเริ่มต้น	79
รูปที่ 2.72 การใช้ฟังก์ชัน SetDlgItemText() กับคอนโทรลอื่น	79
รูปที่ 2.73 ผลการรันการเปลี่ยนแปลงข้อความกับคอนโทรลอื่น	80
รูปที่ 2.74 การกำหนดค่าเริ่มต้นให้อยู่ในสถานะอ่านอย่างเดียว	81
รูปที่ 2.75 รูปแบบของฟังก์ชัน SetReadOnly	81
รูปที่ 2.76 ตัวอย่างการเรียกใช้ฟังก์ชัน SetReadOnly	82
รูปที่ 2.77 การกำหนดเท็กซ์บ็อกซ์ให้เป็นแบบ Password โดยการแก้ไขไอคอนกริซอร์ส	83
รูปที่ 2.78 รูปแบบฟังก์ชัน SetPasswordChar	84
รูปที่ 2.79 การใช้ฟังก์ชัน SetPasswordChar	84
รูปที่ 2.80 การสร้างเท็กซ์บ็อกซ์แบบหลายบรรทัด	85
รูปที่ 2.81 การกำหนดรูปแบบให้เท็กซ์บ็อกซ์แบบหลายบรรทัด	86

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 2.82 การปรับลำดับโฟกัสให้กับแท็บที่บล็อกแบบหลายบรรทัด	86
รูปที่ 2.83 ผลการรันโปรแกรม MultiLine	87
รูปที่ 2.84 การทำให้แท็บที่บล็อกมีความสามารถในการสกรอลล์ข้อความได้	87
รูปที่ 2.85 การยกเลิกลำดับ Tab Order	88
รูปที่ 2.86 การนำข้อมูลในแท็บที่บล็อกมาเก็บลงตัวแปร	90
รูปที่ 2.87 การกำหนดข้อความที่ต้องการให้แสดงที่คอนโทรล	90
รูปที่ 2.88 รูปแบบของฟังก์ชัน UpdateData	90
รูปที่ 2.89 การเพิ่มตัวแปรให้กับคอนโทรล	93
รูปที่ 2.90 ตัวอย่างการใช้ตัวแปรชนิด DDX	93
รูปที่ 2.91 ผลการรันโปรแกรมที่ใช้ตัวแปรชนิด DDX	94
รูปที่ 2.92 ตัวอย่างการใช้ฟังก์ชัน UpdateData(FALSE)	95
รูปที่ 2.93 การสร้างเช็คบ็อกซ์	96
รูปที่ 2.94 การสร้างตัวแปร DDX ให้กับเช็คบ็อกซ์	97
รูปที่ 2.95 การแก้ไขฟังก์ชันให้ให้กับโปรแกรมเช็คบ็อกซ์	98
รูปที่ 2.96 โปรแกรมทดสอบการทำงานของ เรดิโอบ็อกซ์	99
รูปที่ 2.97 การวางกรุปคอนโทรล	101
รูปที่ 2.98 การจัดกลุ่มให้กับเรดิโอบ็อกซ์	102
รูปที่ 2.99 ผลการจัดกลุ่มให้กับเรดิโอบ็อกซ์	103
รูปที่ 2.100 รูปแบบการใช้ฟังก์ชัน AddString	104
รูปที่ 2.101 โปรแกรมทดสอบการทำงานของลิสต์บ็อกซ์	104
รูปที่ 2.102 ผลการทดสอบโปรแกรมลิสต์บ็อกซ์	105
รูปที่ 2.103 การนำการวนรอบมาใช้กับลิสต์บ็อกซ์	105
รูปที่ 2.104 โปรแกรมการวนรอบที่ปรับปรุงใหม่	106
รูปที่ 2.105 ผลการรันโปรแกรมลิสต์บ็อกซ์ที่ปรับปรุงใหม่	107
รูปที่ 2.106 การยกเลิกการ Sort ของลิสต์บ็อกซ์	107
รูปที่ 2.107 ผลการรันเมื่อนำเครื่องหมาย Sort ของลิสต์บ็อกซ์ออก	108
รูปที่ 2.108 โค้ดโปรแกรมในการทดสอบกรณีพิเศษ LBN_DBLCLK	109

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 2.109 ผลการรันโปรแกรมทดสอบการแมปแมสเสจ LBN_DBLCLK	110
รูปที่ 2.110 การสร้างสแตติกคอนโทรล IDC_MESSAGE	111
รูปที่ 2.111 โค้ดโปรแกรมการแมปแมสเสจ LBN_SELCHANGE	111
รูปที่ 2.112 ผลการรันโปรแกรมการแมปแมสเสจ LBN_SELCHANGE	112
รูปที่ 2.113 โค้ดโปรแกรมการใช้กระบวนการ DDX ร่วมกับแมสเสจ LBN_DBLCLK	112
รูปที่ 2.114 โค้ดโปรแกรมการเพิ่มไอเท็มให้กับคอมโบบ็อกซ์	114
รูปที่ 2.115 Dialog Properties	115
รูปที่ 2.116 แท็บ Styles	116
รูปที่ 2.117 แท็บ More Styles	117
รูปที่ 2.118 Extended Styles	118
รูปที่ 2.119 การสร้างเมนูริชอร์ส	121
รูปที่ 2.120 การเปลี่ยนค่า ID ของเมนูริชอร์ส	121
รูปที่ 2.121 การเพิ่มเมนูป๊อปอัพ	122
รูปที่ 2.122 การเพิ่มเมนูย่อย	122
รูปที่ 2.123 การติดตั้งเมนู	123
รูปที่ 2.124 ผลการรันโปรแกรมติดตั้งเมนู	123
รูปที่ 2.125 การใช้ ClassWizard แมปแมสเสจให้กับเมนู	124
รูปที่ 2.126 โปรแกรมทดสอบเมนู	125
รูปที่ 2.127 ผลการรันโปรแกรมทดสอบเมนู	125
รูปที่ 2.128 โปรแกรมทดสอบเมนูป๊อปอัพ	127
รูปที่ 2.129 ผลการรันโปรแกรมทดสอบเมนูป๊อปอัพ	128
รูปที่ 2.130 โปรแกรมแสดงไดอะล็อก	129
รูปที่ 2.131 การวางเท็กซ์บ็อกซ์และสปินคอนโทรล	130
รูปที่ 2.132 การกำหนดให้เท็กซ์บ็อกซ์ร่วมกับสปินคอนโทรล	131
รูปที่ 2.133 โปรแกรมทดสอบสปินคอนโทรล	131
รูปที่ 2.134 ผลการรันโปรแกรมสปินคอนโทรล	132
รูปที่ 2.135 โปรแกรมตรวจสอบค่าในสปินคอนโทรล	133

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 2.136 ผลการทำงานการตรวจสอบค่าในสปีนคอนโทรล	133
รูปที่ 2.137 โปรแกรมทดสอบสไลเดอร์คอนโทรล	134
รูปที่ 2.138 ผลการรันโปรแกรมทดสอบสไลเดอร์คอนโทรล	135
รูปที่ 2.139 โปรแกรมทดสอบสไลเดอร์คอนโทรลร่วมกับเท็กซ์บ็อกซ์	136
รูปที่ 2.140 ผลการรันโปรแกรมทดสอบสไลเดอร์คอนโทรลร่วมกับเท็กซ์บ็อกซ์	137
รูปที่ 2.141 ตัวอย่างการเกิดแมสเสจบ็อกซ์	137
รูปที่ 2.142 ลักษณะของมาร์โครูปไอคอนต่าง ๆ	139
รูปที่ 2.143 รูปแบบการประกาศออบเจกต์ของคลาส CStdioFile	140
รูปที่ 2.144 ตัวอย่างโปรแกรมการอ่านเท็กซ์ไฟล์	141
รูปที่ 2.145 ผลการรันโปรแกรมการอ่านเท็กซ์ไฟล์	141
รูปที่ 2.146 โปรแกรมทดสอบการบันทึกไฟล์	142
รูปที่ 3.1 ผังงานขั้นตอนการออกแบบโปรแกรม	144
รูปที่ 3.2 ผังงานโครงสร้างของโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0	145
รูปที่ 3.3 ผังงานโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0	146
รูปที่ 3.4 ผังงานของบทเรียน	147
รูปที่ 3.5 ผังงานแบบทดสอบ	148
รูปที่ 3.6 ผังงานแบบทดสอบรวม	149
รูปที่ 3.7 ไอคอนล็อกของการ Login	150
รูปที่ 3.8 การกำหนดรูปแบบให้ Text Box เป็นแบบ Password	150
รูปที่ 3.9 เมนูหลัก	151
รูปที่ 3.10 แสดงรายละเอียดของบทเรียน และเมนูย่อย	152
รูปที่ 3.11 โปรแกรมสร้างปุ่มกด	153
รูปที่ 3.12 โปรแกรมแสดงคำอธิบาย	154
รูปที่ 3.13 การสร้างตัวแปร DDX ให้กับ Text Box	155
รูปที่ 3.14 การใช้ฟังก์ชัน ShellExecute()	156
รูปที่ 3.15 รูปแบบของแบบทดสอบที่ได้ออกแบบไว้	156
รูปที่ 3.16 ไอคอนล็อกของข้อสอบที่ออกแบบ	157

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 3.17 แบบทดสอบรวมที่ได้ออกแบบ	158
รูปที่ 3.18 ไดอะล็อกของแบบทดสอบรวม	158
รูปที่ 3.19 ปัญหาในการเขียนโค้ดโปรแกรมภาษาไทย	159
รูปที่ 3.20 การเรียกใช้อุปกรณ์ของ โปรแกรม Visual C++ 6.0	160
รูปที่ 3.21 ไดอะล็อก Options	161
รูปที่ 3.22 การปรับแต่งรูปแบบตัวอักษรให้สามารถแสดงภาษาไทยได้	161
รูปที่ 3.23 ปัญหาในการแสดงผลเป็นภาษาไทย	162
รูปที่ 3.24 การแก้ไขภาษา	162
รูปที่ 3.25 ผลการรันเมื่อเปลี่ยนภาษาที่ใช้แสดง	163
รูปที่ 3.26 การแมปแมสเสจให้ปุ่มกดที่ใช้ในโครงการ	163
รูปที่ 4.1 การลงทะเบียน	164
รูปที่ 4.2 การลงทะเบียนครั้งแรก	165
รูปที่ 4.3 ไดอะล็อกเตือนการใส่ชื่อและรหัสผ่าน	165
รูปที่ 4.4 ไดอะล็อกเตือนการใส่ชื่อซ้ำกับที่มีอยู่แล้วในระบบ	166
รูปที่ 4.5 การยืนยันรหัสผ่าน	166
รูปที่ 4.6 ไดอะล็อกเตือนการใส่รหัส	166
รูปที่ 4.7 คำแนะนำของโปรแกรม	167
รูปที่ 4.8 การเข้าสู่บทเรียนของผู้ที่ผ่านการลงทะเบียนมาแล้ว	167
รูปที่ 4.9 เมนูหลัก	168
รูปที่ 4.10 รายละเอียดและเมนูย่อยของบทเรียนที่ 1	169
รูปที่ 4.11 รายละเอียดและเมนูย่อยของบทเรียนที่ 2	170
รูปที่ 4.12 รายละเอียดและเมนูย่อยของบทเรียนที่ 3	171
รูปที่ 4.13 รายละเอียดและเมนูย่อยของบทเรียนที่ 4	172
รูปที่ 4.14 รายละเอียดและเมนูย่อยของบทเรียนที่ 5	173
รูปที่ 4.15 รายละเอียดและเมนูย่อยของบทเรียนที่ 6	174
รูปที่ 4.16 รายละเอียดและเมนูย่อยของบทเรียนที่ 7	175
รูปที่ 4.17 รายละเอียดและเมนูย่อยของบทเรียนที่ 8	176

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 4.18 รายละเอียดและเมนูย่อยของบทเรียนที่ 9	177
รูปที่ 4.19 เมื่อโปรแกรมคัดลอกไฟล์ตัวอย่างเสร็จสมบูรณ์	178
รูปที่ 4.20 ไคอะล็อกสอบถามความต้องการในการทำแบบทดสอบ	179
รูปที่ 4.21 ไคอะล็อกแบบทดสอบ	179
รูปที่ 4.22 ผลการทดสอบท้ายบทเรียน	180
รูปที่ 4.23 คะแนนสอบในแต่ละบท	180
รูปที่ 4.24 ไคอะล็อกถามความต้องการในการออกจากแบบทดสอบ	181
รูปที่ 4.25 ไคอะล็อกรายงานผลการทำข้อสอบ	181
รูปที่ 4.26 ไคอะล็อกถามความต้องการในการทำแบบทดสอบ	181
รูปที่ 4.27 รายละเอียดในการทำข้อสอบ	182
รูปที่ 4.28 แบบทดสอบรวม	182
รูปที่ 4.29 ผลการทำแบบทดสอบรวม	183
รูปที่ 4.30 ไคอะล็อกเตือนผู้เรียน	183
รูปที่ 4.31 ไคอะล็อกถามความต้องการในการออกจากแบบทดสอบ	183
รูปที่ 4.32 ไคอะล็อกรายงานผลการทำข้อสอบ	184

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

การจัดการศึกษาที่มีครูเป็นผู้นำ ปัญหาที่เกิดขึ้นด้านทรัพยากรประเด็นหนึ่ง คือการขาดแคลนบุคลากร โดยเฉพาะทางด้านวิทยาศาสตร์และเทคโนโลยีจึงจำเป็นที่จะต้องจัดหาสื่อหรือวิธีสอนที่ช่วยลดบทบาทของครูลง ซึ่งได้มีการนำเทคนิคและสื่อการสอนมาประยุกต์เข้าด้วยกัน เพื่อให้สอดคล้องกับความแตกต่างระหว่างบุคคลของผู้เรียน เช่น บทเรียนโปรแกรมชุดการเรียนการสอน การจัดตารางเรียนแบบยืดหยุ่น การใช้คอมพิวเตอร์ช่วยสอน เป็นต้น เทคนิคและสื่อการสอนที่ได้รับความสนใจเป็นอย่างมากคือ การใช้คอมพิวเตอร์ช่วยสอน

เมื่อมีการนำคอมพิวเตอร์เข้ามาช่วยในการสอนในลักษณะของคอมพิวเตอร์ช่วยสอน สิ่งก็ตามมาคือ ความต้องการในการใช้คอมพิวเตอร์เพิ่มสูงขึ้น เป็นผลให้คอมพิวเตอร์ไม่เพียงพอกับความต้องการในการใช้งาน ซึ่งเป็นผลมาจากงบประมาณทางการศึกษาที่มีอยู่อย่างจำกัด ดังนั้น ถ้าต้องการนำคอมพิวเตอร์เข้ามาช่วยในการสอนจะต้องทำอย่างไร ให้ความต้องการใช้ทางด้านปริมาณคอมพิวเตอร์ลดลง

การจัดการเรียนที่นับว่าเป็นหน่วยเล็กที่สุดของปฏิสัมพันธ์ระหว่างบุคคลหรือการเรียนเป็นคู่ นั่นคือ การที่ผู้เรียนสองคนกระทำกิจกรรมการเรียนรู้ร่วมกันจะทำให้ผู้เรียนได้มีโอกาสพัฒนาการสื่อสารกับบุคคลอื่น และมีโอกาสที่ปรับเปลี่ยนมาเป็นบทบาทของผู้สอนให้กับคู่ของตนทำให้การเรียนมีความเข้มข้นสูง

ในปัจจุบัน ระบบปฏิบัติการวินโดวส์ได้เข้ามามีบทบาทอย่างแพร่หลายกับเครื่อง PC และคาดว่าจะมาแทนที่ระบบปฏิบัติการ DOS ได้อย่างสมบูรณ์ ในอนาคตอันใกล้นี้ ดังนั้นในการเขียนโปรแกรมต่างๆบนวินโดวส์จึงเป็นสิ่งสำคัญอย่างหนึ่ง แต่การเขียนโปรแกรมบนระบบปฏิบัติการวินโดวส์ทำได้ไม่ถนัด ดังนั้นจึงมีการผลิตโปรแกรม Visual C++ 6.0 นี้ไม่ใช่เรื่องง่ายสำหรับผู้เริ่มต้น ทั้งยังมีหนังสือให้อ่านประกอบไม่มาก และไม่สามารถมองเห็นภาพได้ในการทำงาน ดังนั้นผู้จัดทำจึงมีความคิดที่จะทำโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0 ขึ้นมาเพื่อให้ผู้ที่ต้องการศึกษาโปรแกรมสามารถนำไปใช้เรียนด้วยตนเองได้ นอกจากนี้ยังแก้ปัญหาในบุคลากรมีจำนวนน้อยได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ขีดความสามารถของโครงการ

-1. โครงการนี้มีบทเรียน 9 บท คือ

- 1) บทที่ 1 เริ่มต้นการใช้งาน
- 2) บทที่ 2 ข้อมูลและการกำหนดข้อมูล
- 3) บทที่ 3 คำสั่งในการควบคุม
- 4) บทที่ 4 อาเรย์และพอยเตอร์
- 5) บทที่ 5 การใช้แบบฟอร์ม และเครื่องมือใน โปรแกรม Visual C++ 6.0
- 6) บทที่ 6 การใช้งานกระบวนการและฟังก์ชัน
- 7) บทที่ 7 ไฟล์
- 8) บทที่ 8 การเชื่อมโยงและฝังออบเจกต์
- 9) บทที่ 9 การสร้างโปรแกรมสมบูรณ์แบบสำหรับดำเนินงานกับระบบปฏิบัติการบนวินโดวส์

2. โครงการสร้างจากโปรแกรม Visual C++ 6.0

3. สามารถใช้งานบนระบบปฏิบัติการ Windows 95 และ Windows 98

4. สามารถใช้เมาส์และคีย์บอร์ดได้

5. บทเรียนแสดงภาพและเสียงในระบบมัลติมีเดีย

6. มีแบบทดสอบท้ายบทเรียน

7. มีตัวอย่างโปรแกรมให้ทดลองศึกษา

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปฏิญญาฉบับนี้แบ่งออกเป็นบทต่างๆ เพื่อสะดวกต่อการศึกษาและทำความเข้าใจ ในแต่ละบทจะประกอบด้วยเนื้อหาดังต่อไปนี้

บทที่ 2 ทฤษฎีและหลักการประกอบด้วยเนื้อหาดังนี้คือ ความหมายของ CAI, คุณสมบัติของ CAI, การออกแบบบทเรียน CAI, การออกแบบโปรแกรมช่วยสอนด้วยคอมพิวเตอร์, ประโยชน์ของโปรแกรมช่วยสอนสำหรับผู้เรียน, ประโยชน์ของโปรแกรมช่วยสอนสำหรับผู้สอน, การพัฒนาของ CAI, Multimedia, แนวความคิดพื้นฐานของ CAI, ทฤษฎีเกี่ยวกับโปรแกรม Visual C++ 6.0 และการใช้งาน โปรแกรม Visual C++ 6.0

บทที่ 3 การออกแบบ การสร้าง และการทำงาน กล่าวถึงเนื้อหาเกี่ยวกับ เครื่องมือที่ใช้ในการออกแบบโปรแกรมช่วยสอน, ขั้นตอนการออกแบบโปรแกรม, โครงสร้างของโปรแกรม, โครงเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สร้างของโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0, บทเรียน, แบบทดสอบ, แบบทดสอบรวม และการสร้างโปรแกรม

บทที่ 4 การทดลอง และผลการทดลอง ประกอบด้วย ความต้องการของโปรแกรม, การเข้าสู่โปรแกรม, การลงทะเบียนเพื่อเข้าสู่บทเรียน, เมนูหลัก, การเลือกบทเรียน, การเข้าสู่บทเรียน, การคัดลอกตัวอย่างโปรแกรม, แบบทดสอบท้ายบทเรียน, แบบทดสอบรวม และผลการตอบแบบสอบถาม

บทที่ 5 บทสรุป ปัญหา แนวทางการแก้ไข และพัฒนา ประกอบไปด้วย สรุปผลของโครงการ, ปัญหาที่พบในการทำโครงการ, การแก้ปัญหา, แนวทางในการพัฒนาสรุปผลของโครงการ, รายงานการวิจัย, ผลการวิจัยแบบสอบถาม และอภิปรายผลการวิจัย

1. ภาคผนวก ก แบบสอบถามประสิทธิภาพการใช้สื่อการสอน



บทที่ 2

ทฤษฎีและหลักการ

2.1 ความหมายของ CAI

CAI (Computer Assisted Instruction) เป็นศัพท์เดิมที่นิยมใช้กันในสหรัฐอเมริกา มีความหมายว่า การสอนโดยใช้คอมพิวเตอร์เป็นเครื่องช่วยสอน แต่ในปัจจุบันมีผู้นิยมใช้คำว่า (Computer-Base Teaching หรือ Computer-Based Teaching) มากกว่า ซึ่งหมายความว่า การสอนหรือการฝึกอบรมโดยใช้คอมพิวเตอร์เป็นหลัก นั่นคือจะเป็นการเน้นการฝึกในด้านต่าง ๆ แทน เช่น ฝึกการซ่อมเครื่องจักร ฝึกการซ่อมวิทยุการฝึกโปรแกรม ฯลฯ นอกจากนี้ในอเมริกาก็ยังมีคำที่นิยมใช้กันอีกคำหนึ่งคือ CMI (Computer Managed Instruction) หมายถึง การสอนโดยใช้คอมพิวเตอร์ช่วยจัดการให้ ส่วนในยุโรป มักใช้คำที่แตกต่างจากอเมริกา คำที่นิยมใช้กันมากในปัจจุบันคือ CBE (Computer-Based Education) หมายถึง การศึกษาโดยอาศัยคอมพิวเตอร์เป็นหลัก นอกจากนี้ยังมีคำว่า CAL (Computer Assisted Learning) และ CML (Computer Managed Learning) นั่นคือเปลี่ยนตัวสุดท้ายจากการสอน (Instruction) เป็นการเรียน (Learning) สำหรับในประเทศไทย มักใช้คำว่า CBI (Computer-Base Instruction) มีความหมายว่า โปรแกรมช่วยสอนที่สามารถใช้ได้ด้วยตนเองโดยไม่จำเป็นต้องมีอาจารย์ผู้สอนคอยชี้แนะ

2.2 คุณสมบัติของ CAI

การใช้งานคอมพิวเตอร์ช่วยสอน ส่วนใหญ่จะหนักไปทางการเรียนด้วยตนเองมากกว่า แม้จะซื้อบทเรียนช่วยสอนก็ตาม กล่าวคือ ผู้เรียนจะเป็นผู้ใช้บทเรียน CAI หรือผู้เข้าฝึกอบรมจะใช้เป็นบทเรียน CBT แนวคิดของ CAI เกิดขึ้นจากนักเทคโนโลยีทางการศึกษาที่ประยุกต์เข้ากับการใช้คอมพิวเตอร์เพื่อการศึกษา โดยแท้จริงแล้ว พื้นฐานของ CAI ก็คือ เครื่องช่วยสอน (Teaching Machine) การมีเครื่องช่วยสอนทำให้ต้องมีโปรแกรมที่เป็นเนื้อหาแบบฝึกหัด และข้อทดสอบที่จะต้องใช้กับเครื่องช่วยสอน ซึ่งก่อนหน้านี้นี้จะมีการประยุกต์ใช้เทคโนโลยีคอมพิวเตอร์ การใช้บทเรียนสำเร็จรูปต่างๆ เช่น บทเรียนโปรแกรม (Program Instruction) บทเรียนโมดูล (Module Instruction)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดการเรียนการสอนสำเร็จรูป (IMP Instruction Package) เป็นต้น โดยเป็นความพยายามที่จะหาวิธีที่จะทำให้ผู้เรียนสามารถเรียนได้ด้วยตนเองตามความสามารถของตนเอง จะใช้เวลามากน้อยต่างกันอย่างไรไม่ว่าจะเกิดการพัฒนบทเรียนสำเร็จรูปเหล่านี้ขึ้น โดยแทนที่จะใช้เครื่องช่วยสอนเป็นตัวเสนอเนื้อหา ก็ใช้หนังสือ (Programmed Test) เป็นตัวเสนอเนื้อหา โดยออกแบบวิธีเสนอเนื้อหาให้สามารถดึงดูดความสนใจของผู้เรียน ใช้เทคนิคของการเสริมแรงและหลักการทางจิตวิทยา การเรียนรู้หลายๆอย่างมาประกอบกันเป็นระบบ

อย่างไรก็ตาม จุดอ่อนของบทเรียนสำเร็จรูปเหล่านี้คือ ความน่าเบื่อ ซึ่งเกิดจากความจำกัดของกิจกรรม ความจำกัดของสิ่งที่นำมาใช้ ความจำเจที่เกิดจากการอ่านเพียงอย่างเดียว การที่ต้องเปิดหนังสือกลับไปกลับมา ความจำเจที่ต้องใช้ประสาทตาอย่างเดียว เมื่อเทคโนโลยีคอมพิวเตอร์พัฒนาขึ้น ทำให้นักการศึกษาหันไปมองหาวิธีขจัดปัญหาต่างๆดังกล่าว โดยการใช้คอมพิวเตอร์เป็นตัวนำเสนอเนื้อหาแทนบทเรียนสำเร็จรูป ซึ่งใช้คอมพิวเตอร์เป็นตัวเสนอเนื้อหาทำให้ได้เปรียบบทเรียนสำเร็จรูปในด้านต่างๆ เหล่านี้

- 1) เสนอเนื้อหาได้รวดเร็วฉับไว แทนการเปิดหนังสือทีละหน้า
- 2) สามารถเสนอภาพเคลื่อนไหวได้
- 3) มีเสียงประกอบ ทำให้เกิดความน่าสนใจ
- 4) สามารถเก็บข้อมูลเนื้อหาได้มากกว่าหนังสือ
- 5) ผู้เรียนมีปฏิสัมพันธ์กับบทเรียนอย่างแท้จริง คือ มีการโต้ตอบระหว่างบทเรียนกับผู้เรียนได้
- 6) สามารถนำติดตัวไปเรียนในสถานที่ต่างๆที่มีเครื่องคอมพิวเตอร์
- 7) เหมาะสำหรับการเรียนการสอนผ่านการสื่อสาร

2.3 การออกแบบบทเรียน CAI

บทเรียน CAI หรือคอมพิวเตอร์ช่วยสอน การสอนมีหลายรูปแบบด้วยกัน ซึ่งการแบ่งรูปแบบหรือประเภทของบทเรียนช่วยสอนด้วยคอมพิวเตอร์ จะขึ้นอยู่กับความคิดเห็นของผู้เชี่ยวชาญด้านนี้ ที่พัฒนาขึ้นเพื่อช่วยเพิ่มประสิทธิภาพการสอนและการรับรู้ของผู้เรียน โดยสรุปมี 5 รูปแบบด้วยกัน คือ

2.3.1 บทเรียน CAI แบบศึกษาเนื้อหาใหม่

บทเรียน CAI แบบศึกษาเนื้อหาใหม่ มีจุดมุ่งหมายที่จะสอนเนื้อหาของหลักการหรือวิชาให้มากที่สุด คือ จะให้คอมพิวเตอร์มีบทบาทเป็นพระเอก โปรแกรม CAI ประเภทนี้ ความจริงเหมือนเอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำราชนิดหนึ่งที่เรียกว่า Programmed Textbook ซึ่งแบ่งข้อความในตำราเป็นกรอบ แต่ละกรอบอธิบายเนื้อหาหนึ่งย่อ แล้วก็ตั้งคำถาม ถ้ามีการตอบผิด ก็จะมีการแนะนำให้พลิกไปอ่านเนื้อหาเพิ่มเติมในหน้าอื่นๆ เป็นต้น โปรแกรม CAI แบบนี้ ทำหน้าที่คล้ายกันแต่ดีกว่า เพราะคอมพิวเตอร์สามารถเลือกข้อความต่างๆมาแสดงให้ดูได้ทันที

บทเรียน CAI แบบศึกษาเนื้อหาใหม่ มี 2 รูปแบบ คือ ชนิดแสดงคำอธิบายต่อเนื่องไปเรื่อยๆ ไม่กระโดดไปหยิบยกเนื้อหาและคำอธิบายอื่นมาเสนอ เรียกว่า ชนิด Linear Tutorial กับอีกรูปแบบคือ ชนิดสามารถเลือกบทเรียนต่างๆได้ตามต้องการ เรียกว่า Branching Tutorial ซึ่งจะเขียนได้ยากกว่า และความสำเร็จของโปรแกรมขึ้นอยู่กับความสามารถของครูที่จะกำหนดเนื้อหาให้เหมาะสม

ดังนั้นผู้ที่ทำโปรแกรมแบบนี้ จะต้องเป็นคนที่สอนวิชานั้นเป็นประจำจนชำนาญ รู้ว่าตรงไหนควรเน้นอะไร และควรมีลำดับเนื้อหาอย่างไร

บทเรียนประเภทแรกนี้เป็นรูปแบบของบทเรียนช่วยสอน ด้วยคอมพิวเตอร์ที่มีผู้พัฒนาจำนวนมากที่สุด มากกว่า 80% ของบทเรียนคอมพิวเตอร์ช่วยสอนทั่วโลกจะเป็นประเภทนี้ เนื่องจากมีพื้นฐานการพัฒนาขึ้นจากความเชื่อที่ว่าคอมพิวเตอร์น่าจะเป็นสื่อประเภทอุปกรณ์ที่ช่วยให้การเรียนรู้มีประสิทธิภาพใกล้เคียงกับการเรียนในชั้นเรียน สรุปคือน่าจะใช้แทนครูได้หลายๆวิชา แนวความคิดตรงนี้มีพื้นฐานในมุมมองที่ว่า การเรียนการสอนนั้น ไม่ได้จำกัดอยู่แต่ในโรงเรียนประถม มัธยมหรืออุดมศึกษา แต่ยังขยายกว้างไปถึงการฝึกอบรมในระดับและสาขาอาชีพต่างๆ ซึ่งอาจผสมผสานการสอน การเรียนรู้และการฝึกฝนด้วยตนเองในหลายรูปแบบ และ CAI แบบ Tutorial ก็อาจเป็นวิธีหนึ่งที่เข้าไปมีบทบาทได้

การใช้บทเรียน CAI แบบ Tutorial ในระบบศึกษาปกติ โดยมีพื้นฐานแนวความคิดที่จะใช้สอนแทนครู ทั้งในห้องเรียนและสอนเสริมนอกเวลาเรียนนั้น ยังเป็นปัญหาที่ต้องใช้เวลาวิเคราะห์กันอีกระยะหนึ่ง ประเด็นไม่ได้อยู่ที่จะทำให้จำนวนครูลดลงหรือขาดบทบาทสำคัญในความเป็นครู แต่จะอยู่ที่ความเชื่อ ในส่วนลึกของผู้คนอีกจำนวนมากที่เชื่อว่าไม่มีสื่อชนิดใดในโลกที่จะถ่ายทอดความรู้ ความคิด ทักษะ และทักษะได้ดีเท่ามนุษย์ด้วยกันเอง ซึ่งหมายถึงครูนั่นเอง ปัญหาการใช้บทเรียน CAI แบบ Tutorial เพื่อสอนแทนครูดังกล่าว ยังรวมถึงความพร้อมทางด้านงบประมาณ โครงสร้างของระบบการศึกษา รวมทั้งปัญหาเฉพาะด้านของแต่ละแห่ง แม้ปัญหาจะมีอยู่มากมาย แต่จากความเชื่อในการพัฒนาด้านเทคโนโลยีคอมพิวเตอร์ที่ไม่มีวันสิ้นสุด ทำให้นักคอมพิวเตอร์การศึกษาเชื่อว่ามีความเป็นไปได้ค่อนข้างสูง ในอนาคตที่จะใช้บทเรียน CAI แบบนี้ เพื่อสอนเสริมสอนถึงบททวน หรือเพื่อให้ผู้เรียนศึกษาหาความรู้ล่วงหน้าก่อนการเรียนในชั้นเรียนปกติ ผู้เรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาจเรียนด้วยความสมัครใจหรืออาจเป็น Assignment จากผู้สอนในหรือนอกเวลาเรียนปกติ ตามแต่กรณี

2.3.2 บทเรียน CAI แบบฝึกทบทวน

บทเรียน CAI แบบฝึกทบทวน มุ่งเน้นที่จะพัฒนาความรู้ความเข้าใจในเรื่องใดเรื่องหนึ่ง โดยเฉพาะ เริ่มต้นด้วยการแสดงตัวอย่าง ซึ่งคอมพิวเตอร์จะแก้ปัญหมาให้ดูทีละขั้น จากนั้นจึงหยิบเอาโจทย์มาให้ให้นักเรียนลองแก้ปัญหาคู ปกติโปรแกรมประเภทนี้ จะเก็บสถิติเอาไว้ว่านักเรียนตอบปัญหาถูกกี่ข้อ และสุดท้ายจะแสดงสถิติให้นักเรียนดู บางโปรแกรมอาจบันทึกเวลาไว้ด้วย CAI ประเภทนี้ สร้างได้ง่าย ไม่ซับซ้อน ใช้เวลาไม่มากนัก

บทเรียนช่วยสอนด้วยคอมพิวเตอร์รูปแบบที่สองนี้ เป็นอีกรูปแบบหนึ่งที่มีผู้พัฒนากันมาก รองลงมาจากประเภทแรก ซึ่งออกแบบขึ้นเพื่อทบทวนบทเรียนที่ได้เรียนไปแล้ว รูปแบบจะเป็นการผสมผสาน การทบทวนแนวความคิดหลักและการฝึกฝน ในรูปแบบการทดสอบ บทเรียนที่พบส่วนมากจะเป็นบทเรียนด้านภาษา คณิตศาสตร์และวิทยาศาสตร์ ซึ่งลักษณะของเนื้อหาจะเน้นด้านความรู้เป็นส่วนมาก จึงไม่เน้นส่วนประกอบหลักๆ ของการเรียนรู้ที่จะต้องมียอดประกอบหลายๆ ด้าน เช่น การนำเสนอเนื้อหาอย่างเป็นระบบตามลำดับขั้น การเสริมแรง การตรวจปรับเนื้อหา สื่อการเรียนการสอน กิจกรรมการเรียนการสอน และอื่นๆ แต่จะเน้นที่แบบฝึกหัดมากกว่า ดังนั้นบทเรียนช่วยสอนประเภทนี้จึงมักจะต้องใช้ควบคู่กับ กิจกรรมอย่างอื่น เช่น ใช้ควบคู่กับการเรียนการสอนปกติในห้องเรียน การให้แบบฝึกหัดเพิ่มเติมในการเรียนเสริม เป็นต้น ซึ่งแตกต่างจากรูปแบบแรกที่เป็นรูปแบบที่สมบูรณ์ในตัวเอง สามารถใช้ในการเรียนการสอนได้ทั้งใน และนอกห้องเรียน

2.3.3 บทเรียน CAI แบบสร้างสถานการณ์จำลอง

บทเรียน CAI แบบนี้ จะออกแบบเพื่อสอนเนื้อหาใหม่ หรือใช้เพื่อทบทวนหรือสอนเสริมในสิ่งที่ศึกษาหรือทดลองไปแล้ว โดยเน้นรูปแบบการสร้างสถานการณ์ การจำลองสถานการณ์จริง ลำดับเหตุการณ์ต่างๆ และเนื้อหาอื่นๆ ที่มีลำดับการเปลี่ยนแปลงอย่างต่อเนื่อง โดยเป็นสิ่งที่เข้าใจยาก ไม่สามารถมองเห็นได้ ต้องอาศัยจินตนาการช่วย ชับซ้อนหรืออันตรายที่จะไปศึกษาในเหตุการณ์จริง ตัวอย่างเช่น อวัยวะร่างกายมนุษย์ โครงสร้างของอะตอม การเกิดปฏิกิริยาทางเคมี หลักการหมุนของมอเตอร์ และอื่นๆ ซึ่งไม่ได้จำกัดเฉพาะด้านวิทยาศาสตร์และเทคโนโลยีเท่านั้น แต่ในด้านธุรกิจสังคมก็สามารถประยุกต์ได้ เช่น การสร้างสถานการณ์ซื้อขาย เพื่อเรียนรู้ หรือทบทวนการบวก ลบ คูณ หาร การสร้างสถานการณ์ในรูปแบบของบทบาทสมมุติ เพื่อสอนหรือทบทวนบทเรียน เป็นต้น

2.3.4 บทเรียน CAI แบบเกมส์สอน

บทเรียน CAI ในลักษณะนี้พัฒนาจากแนวคิด และทฤษฎีทางการเสริมแรง หรือบนพื้นฐานการค้นพบว่า ความต้องการในการเรียนรู้ และความคงทนในการจำดีกว่าการเรียนรู้ที่เกิดจากแรงจูงใจภายนอก แนวคิดและทักษะที่ได้เรียนไปแล้ว คล้ายกับแบบ Drill and Practice แต่เปลี่ยนรูปแบบการนำเสนอให้สนุกตื่นเต้นขึ้น โดยมีหลักการพัฒนาว่า บทเรียนแบบเกมส์สอนที่ดีควรต้องท้าทาย กระตุ้นจินตนาการเพื่อฝัน และกระตุ้นความอยากรู้อยากเห็น จึงเหมาะกับผู้เรียนในระดับต่ำๆ มากกว่าระดับสูง ทั้งนี้เนื่องจากผู้เรียนในระดับต่ำ จำเป็นต้องมีการกระตุ้นด้วย สี แสง เสียง ที่ก่อให้เกิดความสนใจ ความอยากรู้อยากเห็นจึงเหมาะกับเนื้อหาทั่วไป

2.3.5 บทเรียน CAI แบบใช้ทดสอบ

บทเรียนช่วยสอนด้วยคอมพิวเตอร์ประเภทนี้ เป็นรูปแบบที่สร้างง่ายกว่าแบบอื่น จุดประสงค์หลักก็เพื่อทดสอบความรู้ความสามารถของผู้เรียน การสอบดังกล่าวอาจเป็นการสอบก่อนเรียน (Pre-Test) หรือหลังการเรียน (Post-Test) ทั้งก่อนและหลังการเรียนแต่การออกแบบหากเป็นโครงสร้างที่ใหญ่ขึ้น ข้อสอบอาจถูกเก็บในรูปของคลังข้อสอบ (Item Bank) เพื่อสะดวกต่อการสุ่มมาใช้ ลักษณะของข้อสอบดังกล่าวจะอยู่ในรูปแบบที่คอมพิวเตอร์สามารถประเมินถูกผิดได้ เช่น แบบเลือกตอบ (Multiple Choice) หรือแบบถูกผิด (True-False) การตั้งคำถามอาจผสมผสานวิธีการสร้างบทเรียน CAI แบบสร้างสถานการณ์จำลองเข้ามาร่วมด้วยก็ได้

ที่กล่าวมาทั้งหมดนี้เป็นการสรุปรูปแบบของ CAI ที่ได้มีการพัฒนาขึ้นมาใช้อย่างคร่าวๆ แต่ละรูปแบบมีข้อดีไปคนละด้าน อย่างไรก็ตาม ถ้าจะกล่าวถึงเทคนิคการออกแบบบทเรียนคอมพิวเตอร์ช่วยสอน นักคอมพิวเตอร์การศึกษาส่วนมากจะนึกถึงบทเรียนแบบศึกษาเนื้อหาใหม่ เพราะโดยหลักการแล้ว บทเรียนแบบนี้จะมีการประยุกต์เทคนิค และหลักการของบทเรียนอื่นๆ ไม่ว่าจะเป็นแบบฝึกทบทวน แบบสร้างสถานการณ์จำลอง แบบเกมส์การศึกษา หรือแบบใช้ทดสอบเข้ามาอยู่ด้วยกันตามความเหมาะสม โดยอยู่ภายใต้พื้นฐานของธรรมชาติของเนื้อหาที่จะสร้าง ทั้งนี้เนื่องจากความยากง่ายของเนื้อหา และระดับของผู้เรียนก็เป็นองค์ประกอบที่สำคัญประการหนึ่งที่จะต้องพิจารณาเลือกรูปแบบของบทเรียนที่จะสร้าง

2.4 การออกแบบโปรแกรมช่วยสอนด้วยคอมพิวเตอร์

การออกแบบโปรแกรมด้วยคอมพิวเตอร์แบบ Tutorials จะยึดหลักการเรียนการสอนเป็นหลักพื้นฐาน โดยจะดัดแปลงมาจากกระบวนการเรียนการสอน 9 ขั้นของ Gagne ตามลำดับขั้น ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 ไร่้ความสนใจ

ก่อนที่จะเริ่มเรียนนั้นมีความจำเป็นอย่างยั้งที่ผู้เรียนควรจะได้รับแรงกระตุ้นและแรงจูงใจที่อยากจะทำเรียน ดังนั้นบทเรียนควรจะเริ่มลักษณะของการใช้ภาพ แสง เสียง หรือการประกอบกันหลายๆอย่างโดยมีสิ่งทีสร้างขึ้นมาั้นเกี่ยวข้องกับเนื้อหา และน่าสนใจ ซึ่งจะมีผลต่อความสนใจจากผู้เรียน และเป็นการเตรียมผู้เรียนให้พร้อมที่จะศึกษาเนื้อหาต่อไป การเตรียมตัวและกระตุ้นผู้เรียนในชั้นแรกนี้เป็นการสร้าง Title ของบทเรียน ข้อสำคัญประการหนึ่งในชั้นนี้ คือการสร้าง Title ควรออกแบบเพื่อให้เป็นการตอบสนองอย่างง่าย เช่น การกด Spacebar หรือการกดคีย์ใดๆ

สิ่งทีต้องพิจารณา เพื่อไร่้ความสนใจของผู้เรียน มีดังนี้

- 1) ใช้กราฟฟิคทีเกี่ยวข้องกับเนื้อหา และกราฟฟิคนั้นควรมีขนาดใหญ่ง่าย และไม่ซับซ้อน
- 2) ใช้ภาพเคลื่อนไหว หรือเทคนิคอื่นๆเข้าช่วยเพื่อแสดงการเคลื่อนไหว แต่ควรสั้นและง่าย
- 3) ควรใช้สีเข้าช่วยโดยเฉพาะสีเขียว แดง น้ำเงิน หรือสีเข้มอื่นๆ ทีตัดกับพื้นชัดเจน
- 4) ใช้เสียงให้สอดคล้องกับกราฟฟิค
- 5) กราฟฟิคควรจะมีค่าบงอภาพจนกว่าผู้เรียนจะกดคีย์
- 6) ในกราฟฟิคดังกล่าวควรบอกชื่อเรื่องบทเรียนไว้ด้วย
- 7) ควรใช้เทคนิคการเขียนกราฟฟิคทีแสดงบงอได้เร็ว
- 8) กราฟฟิคนั้นนอกจากจะเกี่ยวกับเนื้อหาแล้ว ต้องเหมาะสมกับวัยของผู้เรียนด้วย

2.4.2 บอกวัตถุประสงค์

บอกวัตถุประสงค์ของการเรียนในบทเรียนคอมพิวเตอร์นั้นเพื่อให้ผู้เรียนได้รู้ล่วงหน้าถึงประเด็นสำคัญของเนื้อหาแล้วยังเป็นการบอกถึงเค้าโครงเรื่องอีกด้วย การทีผู้เรียนทราบถึงโครงร่างของเนื้อหาอย่างกว้างๆ จะช่วยให้ผู้เรียนสามารถผสมผสานแนวความคิดในรายละเอียด หรือส่วนย่อยของเนื้อหาให้สอดคล้อง และสัมพันธ์กับเนื้อหาส่วนใหญ่ได้ซึ่งมีผลทำให้การเรียนรู้มีประสิทธิภาพขึ้น และนอกจากจะมีผลดังกล่าวแล้ว การวิจัยยังพบว่าผู้เรียนทีทราบวัตถุประสงค์ของการเรียนก่อนบทเรียน จะสามารถจำและเข้าใจในเนื้อหาได้อีกด้วย

การบอกวัตถุประสงค์ทำได้หลายแบบ ตั้งแต่แบบทีเป็นวัตถุประสงค์กว้างๆ จนกระทั่งการบอกวัตถุประสงค์เชิงพฤติกรรม ในการออกแบบบทเรียนคอมพิวเตอร์ช่วยสอนนั้น หลักการทีสำคัญอย่างหนึ่ง คือข้อความทีเสนอบงอ ควรเป็นข้อความทีสั้น ได้ใจความและควรมีส่วนจูงใจผู้

เรียนด้วย ดังนั้นการบอกถึงวัตถุประสงค์ในบทเรียน CAI จึงนิยมใช้ข้อความที่สั้น และโน้มน้าวจิตใจผู้เรียน

สิ่งที่ต้องพิจารณาในการบอกวัตถุประสงค์ มีดังนี้

- 1) ใช้คำสั้นๆ และเข้าใจง่าย
- 2) หลีกเลี่ยงคำที่ยังไม่รู้จัก
- 3) ไม่ควรกำหนดวัตถุประสงค์หลายข้อเกิดไปในเนื้อหาแต่ละส่วน
- 4) ผู้เรียนควรมีโอกาสทราบว่าหลังการเรียนจบแล้วจะนำไปใช้ทำอะไรได้บ้าง
- 5) หากบทเรียนนั้นมีบทเรียนย่อยหลายบท หลังจากทีบอกวัตถุประสงค์กว้างๆ แล้ว อาจจะบอกวัตถุประสงค์อีกครั้งในแต่ละส่วนย่อยได้
- 6) เพื่อให้วัตถุประสงค์น่าสนใจยิ่งขึ้นอาจใช้กราฟที่ง่ายๆ เข้าช่วย เช่น กรอบ, ลูกศร และ รูปทรงเรขาคณิต แต่ไม่ควรใช้การเคลื่อนไหวกับตัวหนังสือ

2.4.3 ทวนความรู้เดิม

ก่อนที่จะให้ความรู้ใหม่แก่ผู้เรียน ซึ่งในส่วนของเนื้อหาและแนวความคิดนั้นๆ ผู้เรียนอาจไม่มีพื้นฐานมาก่อน มีความจำเป็นอย่างยิ่งที่ผู้ออกแบบบทเรียน CAI จะต้องหาวิธีการประเมินผลความรู้เดิมที่จำเป็นก่อนที่จะได้รับความรู้ใหม่ เพื่อเป็นการเตรียมผู้เรียนให้พร้อมที่จะรับความรู้ใหม่

ในขั้นทบทวนความรู้เดิมนี้อาจจำเป็นต้องเป็นการทดสอบเสมอไป หากเป็นบทเรียนที่สร้างขึ้นเป็นชุด บทเรียนที่เรียนต่อๆ กันไปตามลำดับ การทบทวนบทเรียนความรู้เดิมอาจเป็นไปในรูปแบบของการกระตุ้นให้ผู้เรียนคิดย้อนหลังถึงสิ่งที่ได้เรียนมาก่อนหน้านี้ การกระตุ้นดังกล่าวอาจแสดงด้วยคำพูด การเขียน หรือเป็นการผสมผสานกันแล้วแต่ความเหมาะสม จะมากหรือน้อยขึ้นอยู่กับความเหมาะสมของเนื้อหา

สิ่งที่ต้องพิจารณาในการทบทวนบทความรู้เดิม มีดังนี้

- 1) ไม่ควรคาดเดาว่าผู้เรียนมีความรู้พื้นฐานก่อนการศึกษาเนื้อหาใหม่เท่ากัน ควรมีการทดสอบหรือให้ความรู้เพื่อเป็นการทบทวนให้ผู้เรียนพร้อมที่จะรับความรู้ใหม่
- 2) การทบทวนหรือทดสอบควรให้กระชับและตรงตามวัตถุประสงค์มากที่สุด
- 3) ควรเปิดโอกาสให้ผู้เรียนออกจากเนื้อหาหรือแบบทดสอบเพื่อไปศึกษาทบทวนได้
- 4) หากไม่มีการทดสอบความรู้เดิม ผู้เขียนโปรแกรมควรหาทางกระตุ้นให้ผู้เรียนย้อนคิด จะทำให้บทเรียนน่าสนใจขึ้น

2.4.4 การเสนอเนื้อหา

การเสนอภาพที่เกี่ยวข้องกับเนื้อหาที่ประกอบกับคำพูดที่สั้น ง่ายและได้ใจความ เป็นหัวใจสำคัญของบทเรียนช่วยสอนด้วยคอมพิวเตอร์ การใช้ภาพประกอบจะทำให้ผู้เรียน เข้าใจเนื้อหาได้ง่ายขึ้น ซึ่งภาพจะช่วยอธิบายสิ่งที่เป็นนามธรรม

ภาพที่ใช้ในบทเรียนช่วยสอนจำแนกออกได้สองส่วนใหญ่ๆ คือ ภาพนิ่ง (Still Picture) ได้แก่ ภาพลายเส้น ภาพสองมิติ ภาพถ่ายของจริง แผนภาพ กราฟ และอื่นๆ อีกส่วนหนึ่งได้แก่ ภาพเคลื่อนไหว (Motion Picture) เช่น ภาพจากสัญญาณวิดีโอ (Video) ภาพจากสัญญาณดิจิทัลอื่นๆ เช่นจาก Photo CD จาก Laser Disk จากกล้องถ่ายภาพโทรทัศน์ เป็นต้น

ในส่วนของเนื้อหาที่เสนอเป็นคำอ่านหรือคำอธิบายนั้น ในแต่ละกรอบไม่ควรมีมากเกินไป เพราะจะทำให้ผู้เรียนเบื่อกว่าต้องนั่งอ่าน โดยไม่ได้ทำอะไรเลย นอกจากการกดแป้นพิมพ์ และยังทำให้อ่านยากอีกด้วย

2.4.5 ชี้นำทางการเรียนรู้

ตามหลักการเรียนรู้ ผู้เรียนจะจำได้ดี หากมีการจัดระบบการนำเสนอเนื้อหาที่ดี และสัมพันธ์กับประสบการณ์เดิมหรือความรู้เดิม ทฤษฎีบางทฤษฎีได้กล่าวว่าการเรียนรู้ที่กระจำจืด จะเกิดขึ้นได้เมื่อผู้เรียนเกิดการตีความของเนื้อหาใหม่บนพื้นฐานความรู้เดิม รวมกันเป็นความรู้ใหม่ ดังนั้นหน้าที่ของผู้ออกแบบบทเรียน CAI ก็คือ พยายามหาเทคนิคในการที่จะกระตุ้นให้ผู้เรียนนำความรู้เดิมมาใช้ในการศึกษาความรู้ใหม่ นอกจากนั้นยังต้องพยายามหาวิธีทางที่จะทำให้การศึกษาความรู้ใหม่ของผู้เรียนนั้นมีความกระจำเท่าที่จะทำได้

ในบางเนื้อหาผู้ออกแบบบทเรียน CAI อาจใช้หลักของ Guide Discovery ซึ่งหมายถึง การพยายามให้ผู้เรียนคิดหาเหตุผล ค้นคว้าและวิเคราะห์หาคำตอบด้วยตนเองและเทคนิคการให้ตัวอย่าง นอกจากนั้นการใช้คำพูดกระตุ้นให้ผู้เรียนได้คิดก็เป็นเทคนิคอีกอย่างหนึ่งที่จะใช้ได้

2.4.6 กระตุ้นการตอบสนอง

ทฤษฎีการเรียนรู้ที่กล่าวว่า การเรียนรู้จะมีประสิทธิภาพมากน้อยนั้น เกี่ยวข้องโดยตรงกับระดับและขั้นตอนของการประมวลข้อมูล หากผู้เรียนได้มีโอกาสร่วมคิดร่วมกิจกรรม ในส่วนที่เกี่ยวข้องกับเนื้อหา การถาม ตอบ ในด้านของการจำจะดีกว่าการคิดลอกเนื้อหาไปเพียงอย่างเดียว

คอมพิวเตอร์มีข้อได้เปรียบเหนืออุปกรณ์อื่นๆ อย่างเช่น วีดิโอเทป ภาพยนตร์สไลด์ เทป หรือสื่อการสอนอื่นๆ ซึ่งเป็นการสอนแบบ Non-interactive คือ การเรียนจากคอมพิวเตอร์นั้นผู้เรียนสามารถมีกิจกรรมร่วมได้หลายลักษณะ แม้จะเป็นการแสดงความคิดเห็น การโต้ตอบกับเครื่องคอมพิวเตอร์ทำให้ผู้เรียนไม่เบื่อหน่าย

2.4.7 ให้ข้อมูลย้อนกลับ

การค้นคว้าพบว่าบทเรียน CAI นั้นกระตุ้นความสนใจจากผู้เรียนมากขึ้น ถ้าบทเรียนนั้นทำโดยผู้เล่น โดยการบอกจุดหมายที่ชัดเจนและให้ Feedback เพื่อบอกว่าขณะนั้นผู้เรียนอยู่ตรงไหน ห่างจากเป้าหมายเท่าใด

การ Feedback ที่เป็นภาพจะช่วยเร้าความสนใจยิ่งขึ้น โดยเฉพาะอย่างยิ่งถ้าภาพนั้นเกี่ยวข้องกับเนื้อหา

2.4.8 ทดสอบความรู้หลังบทเรียน

บทเรียน CAI จัดเป็นบทเรียนสำเร็จรูปประเภทหนึ่ง การทดสอบความรู้ใหม่ ซึ่งอาจจะเป็นการทดสอบระหว่างเรียน หรือการทดสอบในช่วงท้ายของบทเรียน เรียกว่า Post Test เป็นสิ่งที่จำเป็น การทดสอบดังกล่าวอาจเป็นการเปิดโอกาสให้ผู้เรียนได้ทดสอบตนเอง เพื่อเก็บคะแนนหรือเป็นการวัดผลว่าผู้เรียนผ่านเกณฑ์ เพื่อที่จะได้ศึกษาในบทต่อไป

การทดสอบดังกล่าว นอกจากจะเป็นการประเมินการเรียนแล้ว ยังมีผลในการจำระยะยาวของผู้เรียนด้วย ข้อสอบจึงควรตามเรียงตามวัตถุประสงค์ของบทเรียน

2.4.9 การจำและแนะนำการใช้

ในการเตรียมตัวการสอนสำหรับชั้นเรียนปกติ ตามข้อเสนอแนะของ Gagne นั้น ในขั้นนี้จะเป็นกิจกรรมสรุปเฉพาะประเด็นสำคัญ รวมทั้งข้อเสนอแนะต่างๆ เพื่อให้ผู้เรียนมีโอกาสได้ทบทวน

2.5 ประโยชน์ของโปรแกรมช่วยสอนสำหรับผู้เรียน

- 1) เพื่อใช้เวลาว่างให้เป็นประโยชน์ คือ โปรแกรม CAI ใช้หลักการที่เรียกว่า Individualized Learning หมายความว่า นักเรียนสามารถเรียนได้ช้าหรือเร็วเท่ากับความสามารถของตนเอง ใครที่ฉลาดและเรียนรู้ได้เร็วก็เรียนจบเร็วกว่าคนอื่น ไม่ต้องเสียเวลารอคอยไปด้วยกันทั้งชั้น
- 2) สามารถเขียนพร้อมกันได้ ในกรณีการฝึกอบรมนั้น บางครั้งผู้เรียนมีหลายคนทำให้ไม่พร้อมกัน การฝึกก็ต้องเลื่อนออกไป ทำให้เสียเวลารอคอย
- 3) บทเรียนมีประสิทธิภาพ และมีความเชื่อมั่นสูง การเรียนกับครูนั้น บางครั้งผู้เรียนต่างห้อง จะได้รับความรู้ไม่เท่ากัน ขึ้นอยู่กับอารมณ์ และการเตรียมการสอนของครู
- 4) ลดเวลาการเดินทาง บทเรียน CAI สามารถที่จะเรียนที่บ้านได้ถ้ามีคอมพิวเตอร์ ไม่ต้องเสียเวลาเดินทาง
- 5) สามารถเรียนได้ทั้งกลางวันและกลางคืน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ประโยชน์ของโปรแกรมช่วยสอนสำหรับผู้สอน

- 1) ลดเวลาเตรียมการสอนในระยะยาว ในการเตรียมการสอนนั้นต้องใช้เวลาาน ถึงแม้ว่าจะเป็นวิชาที่เคยสอนมาก่อน
- 2) ปรับปรุงแก้ไขบทเรียนได้เร็ว และง่าย คือ สามารถที่จะเพิ่มเติมเนื้อหาและรายละเอียดต่างๆ เข้าไปได้ง่าย
- 3) สามารถควบคุมผลสัมฤทธิ์ได้ง่าย

2.7 การพัฒนาการของ CAI, Multimedia และ ITS จะเกี่ยวข้องกับระบบการศึกษา

2.7.1 ข้อดีของ CAI มากกว่าแบบสอนธรรมดา

- 1) ใช้งบประมาณจำนวนน้อย, การพัฒนาที่เร็วและมีประสิทธิภาพ, เป็นการเรียนรู้ในระยะเวลาที่สั้น, สะดวกในการเคลื่อนย้ายและการเก็บรักษา, สะดวกและเมื่อเทียบกับสอนแบบธรรมดา ราคาจะต่ำกว่า และสามารถแจกจ่ายไปยังเครือข่ายต่างๆ ได้
- 2) ภาพเคลื่อนไหว ทำได้ง่าย, เร็ว, ง่ายต่อการแก้ไขดัดแปลง และราคาไม่แพงเลยสำหรับการจำหน่าย

2.7.2 ขีดจำกัด

- 1) การเก็บข้อมูล เหมือนกับการพูดโดยเป็นการโต้ตอบทางเดียว
- 2) ไม่เป็นธรรมชาติ การสื่อสารจะใช้คีย์บอร์ด, เมาส์ และตัวหนังสือ

2.7.3 การพัฒนาของ CAI

- 1) เสียงสำหรับการจำและส่วนประกอบทั้งหมดที่รวมกัน
- 2) ความรู้พื้นฐาน
- 3) วิธีการสอนแบบใหม่
- 4) รูปแบบของนักเรียน
- 5) ITS
- 6) มัลติมีเดีย
- 7) ภาพที่สมจริง
- 8) เครื่องบาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.4 ตามแบบ IMS การใช้คอมพิวเตอร์ วาด, บันทึก, การอ่าน

- 1) Wordprocessor และการแก้ไขกราฟฟิก
- 2) สมแกนเนอร์
- 3) กล้องวิดีโอ

2.7.5 ภาพเคลื่อนไหว

- 1) Graphic / Video Editors
- 2) Animators
- 3) กล้องวิดีโอ
- 4) การ์ดจับภาพวิดีโอ
- 5) ตัวแปลง VGA to Video

2.8 แนวความคิดพื้นฐาน ของ CAI

2.8.1 การทำให้คอมพิวเตอร์ทำได้เท่าเทียมกับแบบที่ครูสอน

- 1) การทำโดยไม่ซ้ำแบบกับผู้อื่น
- 2) แก้ไขปรับปรุงการโต้ตอบ
- 3) ต้องคิดด้วยตนเอง

2.8.2 ประสบการณ์พิเศษ

มีการโต้ตอบแบบจับปล้น

2.8.3 โครงสร้าง

- 1) ข้อมูลเป็นปัจจุบัน
- 2) ตั้งคำถาม
- 3) ขอมรับการตอบสนอง
- 4) ผลการประเมิน และเตรียมการโต้ตอบ
- 5) พิจารณาข้อมูลต่อไป
- 6) คำถามต้องเป็นปัจจุบัน

2.8.4 การพัฒนาให้ CAI ดึงดูดความสนใจได้

- 1) สามารถใช้ประสาทสัมผัสได้หลายทาง
- 2) มีการโต้ตอบแบบ 2 ทิศทาง
- 3) ใช้แทนบทบาทของครู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) การสร้างนักเรียนเป็นศูนย์กลาง
- 5) การสอนที่สอนไถ่ได้

2.8.5 ผลกระทบของ CAI ในการสร้างบทเรียน

- 1) เป็นการลงทุนเพื่อหาผลประโยชน์
- 2) ราคาต่ำ
- 3) แรงจูงใจอยู่ในระดับสูง
- 4) ความหมกมุ่นในการใช้ CBT LAB
- 5) กลุ่มที่รวมกัน และการเรียรายบุคคล
- 6) ทำให้นักเรียนอาจใช้ข้อมูลใน CBI มากกว่า
- 7) ประสิทธิภาพของการเรียนจากประสบการณ์ที่เป็นจินตนาการ
- 8) การเห็นภาพที่เป็นจินตนาการ, ความคิดริเริ่มจากตนเองและความคิดแบบทันทีทันใด และความจำที่มาก
- 9) การเปลี่ยนแปลงบทบาทของครูและนักเรียน
- 10) การเปลี่ยนแปลงใน โครงสร้างของจริยศาสตร์
- 11) หลักสูตรเปลี่ยนแปลง
- 12) การวิจัยที่ต้องการในการทางปรัชญาหรือจริยศาสตร์

2.8.6 ขีดจำกัดของ CAI ในการสร้างบทเรียน

- 1) ขีดจำกัดของเสียงในการจำ
- 2) การมองโลกในการดำรงชีวิต
- 3) ประสบการณ์จากการทำด้วยมือ
- 4) ขีดจำกัดของนักเรียน
- 5) ความรู้พื้นฐานที่ไม่เท่ากัน
- 6) ความคิดที่สับสน
- 7) สื่อที่ไม่ธรรมดาในการเรียนการสอน
- 8) นักเรียนใช้ความชำนาญในการเรียน

2.8.7 รูปแบบของการเรียนด้วยคอมพิวเตอร์

- 1) ความสนใจของนักเรียนกับภาพจินตนาการ
- 2) สนับสนุนกิจกรรมกับนักเรียนหรือไม่
- 3) ช่วยเหลือความต้องการของนักเรียนหรือไม่
- 4) การวางแผนการสอนแบบเดี่ยวหรือแบบกลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนเวลาสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 5) สามารถเรียนด้วยตนเองหรือเป็นการสอนแบบทางไกล
- 6) บทบาทของครูในรูปแบบนี้ คืออะไร

2.8.8 โครงสร้างเกี่ยวกับการสอน

- 1) ข้อมูลเป็นปัจจุบัน, ตัวอักษร, กราฟฟิกส์, รูปภาพ, เสียง และวีดีโอ
- 2) ตั้งคำถาม
- 3) ยอมรับการตอบสนอง
- 4) วิเคราะห์การตอบสนอง
- 5) วิธีการตอบสนองกลับ
- 6) การเชื่อมต่อกับจากต่อไป

2.8.9 การใช้

- 1) ใช้แนะนำบทเรียนใหม่, แนวความคิด และการสร้าง ฯลฯ
- 2) นอกจากนี้เป็นการพัฒนารูปแบบที่เปลี่ยนได้ และความสามารถของระบบ

2.8.10 แบบฝึกหัด

- 1) กำหนดแบบฝึกหัดให้กับนักเรียน
- 2) คำแนะนำ และความช่วยเหลือในผลเฉลยของปัญหา
- 3) การประเมินผลของนักเรียน
- 4) การทดสอบตนเอง
- 5) เวลาในการทำ

2.9 ทฤษฎีเกี่ยวกับโปรแกรม Visual C++ 6.0

ระบบปฏิบัติการวินโดวส์เป็นระบบปฏิบัติการ แบบกราฟฟิกที่อำนวยความสะดวกในการใช้งานได้เป็นอย่างมาก เพราะรูปแบบการใช้งานจะเป็นแบบหน้าต่าง และมี User Interface เพื่อใช้ในการติดต่อกับผู้ใช้ในรูปแบบกราฟฟิก

ระบบปฏิบัติการวินโดวส์ได้ถูกพัฒนาจากเวอร์ชัน 3 มาเป็นระบบปฏิบัติการ Windows 98 เพราะเป็นระบบปฏิบัติการที่เพียบพร้อมด้วยความสามารถครบครันทั้งในด้าน การใช้งานที่ง่ายกว่า ระบบปฏิบัติการวินโดวส์เวอร์ชันเก่า, การทำงานด้านเอกสาร และทำงานด้านระบบเครือข่ายต่างๆ และที่สำคัญก็คือ ระบบปฏิบัติการ Windows 95 กำลังได้รับความนิยมกันอยู่เป็นอย่างมากในขณะนี้

สิ่งหนึ่งที่ทำให้ระบบปฏิบัติการวินโดวส์ใช้งานได้ง่ายก็คือ รูปแบบการติดต่อกับผู้ใช้ หรือ User Interface ของวินโดวส์นั่นเอง โปรแกรมต่างๆ ที่ออกแบบขึ้นมาเพื่อให้ทำงานบนระบบปฏิบัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวินโดวส์ก็จะต้องใช้รูปแบบการ Interface ของวินโดวส์ การพัฒนาโปรแกรมบนระบบปฏิบัติการวินโดวส์จึงแตกต่างจากการพัฒนาโปรแกรมบนคอสมอส เพราะการรับข้อมูลและการแสดงผลให้กับผู้ใช้ไม่อยู่ที่บรรทัดคำสั่ง หรือคอมมานด์ไลน์ (Command Line) แบบในคอสมอสอีกแล้วแต่จะอยู่ที่คอนโทรลและอุปกรณ์ต่างๆ ภายในวินโดวส์นั่นเอง เช่น ปุ่มกด, ช่องรับข้อความ และเช็คบ็อกซ์ เป็นต้น

ยกตัวอย่างเช่น หากเขียนโปรแกรมบนคอสมอสก่อน จะต้องการรับค่าจากผู้ใช้ เช่น ชื่อและนามสกุล ถ้าเป็นภาษาปาสคาลจะต้องใช้คำสั่ง Readln รับค่าเข้ามา โดยเคอร์เซอร์จะรอรับอยู่บนจอภาพ แต่ถ้าเขียนโปรแกรมในสภาวะแวดล้อม ของวินโดวส์จะต้องพิมพ์มันลงในเท็กซ์บ็อกซ์ (ช่องรับข้อความ) นั่นเอง

เพราะฉะนั้น ในการพัฒนาโปรแกรมบนระบบปฏิบัติการวินโดวส์สิ่งหนึ่งที่จะต้องศึกษาให้ดี และควรศึกษาเป็นสิ่งแรกก็คือ การสร้างและควบคุมระบบ Interface ของวินโดวส์ นั่นคือ จะต้องศึกษาการสร้างวินโดวส์, การสร้างปุ่มกด และเท็กซ์บ็อกซ์และลิสต์บ็อกซ์ เพราะสิ่งเหล่านี้ เป็นอุปกรณ์ พื้นฐานในการติดต่อกับผู้ใช้ และในการทำงานเลยทีเดียว ถึงแม้ว่าจะเข้าใจวิธีการจัดการระบบฐานข้อมูลโปรแกรม หรือรู้จักวิธีการจัดเก็บข้อมูลลงไฟล์อย่างคีย์ออกเยียม แต่ถ้าไม่รู้วิธีการสร้างและการใช้งาน Interface ต่างๆ เหล่านี้ ก็ไม่สามารถที่จะรับข้อมูลหรือแสดงผลในโปรแกรม ข้อมูลได้

ในการพัฒนาโปรแกรมบนระบบปฏิบัติการวินโดวส์ในยุคแรกๆ จะใช้เครื่องมือที่ เรียกว่า SDK (Software Development Kit) ซึ่งจะเน้นชุดพัฒนาโปรแกรมบนระบบปฏิบัติการวินโดวส์ แต่ในปัจจุบันนี้ได้มีการสร้างเครื่องมือที่ช่วยในการพัฒนาโปรแกรมบนระบบปฏิบัติการวินโดวส์ขึ้นมาหลาย เช่น Microsoft Visual Basic, Visual C++, Delphi หรือ Power Builder เป็นต้น ซึ่งโดยมากจะเป็นโปรแกรมประเภท Visual นั่นคือ รูปแบบการพัฒนาโปรแกรมจะอยู่ในลักษณะของ WYSIWYG

ในปัจจุบันโปรแกรม Visual C++ 6.0 ได้รับการพัฒนามาจนถึงเวอร์ชัน 6.0 และเป็นเครื่องมือตัวที่ใช้ในการพัฒนาโปรแกรมในชุดพัฒนาของไมโครซอฟท์ที่ชื่อ Visual Studio 98

2.9.1 Visual Studio 98

Visual Studio 98 เป็นชุดเครื่องมือที่ใช้ในการพัฒนาโปรแกรมที่สมบูรณ์แบบในปัจจุบันพร้อมสรรพด้วยความสามารถหลายๆ ด้าน Visual Studio 98 ได้รับความนิยมเป็นอย่างมากในหมู่นักพัฒนาโปรแกรม รวมทั้งในหมู่นักไทยด้วยกัน เพราะความสามารถและคุณลักษณะใหม่ๆ ที่ได้เพิ่มเข้าไปทำให้การพัฒนา โปรแกรมง่ายขึ้น ไม่ว่าจะ เป็น บน Window, บน Internet, ภาษา Java, Active Server Page และ ActiveX เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2 Visual Studio 98 ประกอบด้วยเครื่องมือในการพัฒนาโปรแกรมมี ดังนี้

- 1) Microsoft Visual C++ 6.0
- 2) Microsoft Visual Basic 6.0
- 3) Microsoft Visual FoxPro 6.0
- 4) Microsoft Visual InterDev 6.0
- 5) Microsoft Visual J++ 6.0
- 6) Microsoft Visual SourceSafe 6.0

และ Microsoft Developer Network Library ซึ่งจะเป็นไลบรารีระบบช่วยเหลือของโปรแกรมต่างๆ ในชุด Visual Studio 98 นี้ หากเคยพัฒนาโปรแกรมด้วย Visual C++ เวอร์ชันที่ผ่านๆ มา เช่น Visual C++ 5.0 ซึ่งเป็นผลิตภัณฑ์ในปี ค.ศ. 1997 ที่อยู่ภายใต้ชุดเครื่องมือ Visual Studio 97 ก็สามารถใช้งาน Visual Studio 98 ได้ง่าย เพราะ IDE (Integrated Development Environment) ของ Visual Studio 98 มีรูปแบบการใช้งานที่คล้ายๆ กัน

2.9.3 ประวัติของ โปรแกรม Visual C++ 6.0 และ MFC

โปรแกรม Visual C++ 6.0 ได้รับการพัฒนาขึ้นมาจาก Microsoft C/C++ ที่ทำงานบนระบบปฏิบัติการวินโดวส์ได้อย่างเต็มที่ รองรับการพัฒนาการโปรแกรมบนวินโดวส์โดยมี MFC (Microsoft Foundation Class) เป็นไลบรารีที่ช่วยอำนวยความสะดวกในการพัฒนาโปรแกรมบนวินโดวส์

ในยุคที่ MFC ยังไม่ถือกำเนิด เมื่อจะพัฒนาโปรแกรมบนระบบปฏิบัติการวินโดวส์จะต้องใช้ SDK (Software Development Kit) และคอมไพเลอร์ภาษา C เช่น Microsoft C++, Borland C++ ช่วยในการเขียนโปรแกรม โค้ดโปรแกรมที่เขียนขึ้นด้วย SDK นี้จะค่อนข้างซับซ้อน และสร้างความลำบากในการศึกษาทำความเข้าใจ จึงได้มีการสร้างคลาสขึ้นมาหนึ่ง เขียนขึ้นโดยใช้โครงสร้างของ OPP (Object Oriented Programming) ด้วยภาษา C++ ชื่อว่า MFC หรือ Microsoft Foundation Class ใช้สำหรับอำนวยความสะดวกในการเขียนโปรแกรมบนวินโดวส์ โดยเฉพาะ โค้ดโปรแกรมที่เขียนโดยใช้ MFC นั้น จะมีขนาดเล็กและไม่ซับซ้อน ทำให้การพัฒนาโปรแกรมบนวินโดวส์สามารถทำได้ง่ายกว่าใช้ SDK

จากความยืดหยุ่นและประสิทธิภาพของ MFC นี้ทำให้ MFC ได้รับความนิยมแพร่หลายมากยิ่งขึ้น และได้ถูกรวมเป็นไลบรารีหนึ่งของ Microsoft C/C++ เวอร์ชัน 7 และคอมไพเลอร์ตัวอื่นๆ เช่น Symantec C++ เป็นต้น

MFC นั้นเป็นส่วนหนึ่งของ โปรแกรม Visual C++ 6.0 เพราะ โปรแกรม Visual C++ 6.0 เป็นเครื่องมือในการพัฒนาโปรแกรมบนวินโดวส์ที่มี MFC เป็นไลบรารีที่ช่วยอำนวยความสะดวกแทนการใช้ SDK ซึ่งทำให้ MFC ก็ได้รับการพัฒนาให้มี ประสิทธิภาพสูงขึ้นไปพร้อมกับ โปรแกรม Visual C++ จนถึงทุกวันนี้

ตารางที่ 2.1 ประวัติการพัฒนาของ MFC และโปรแกรม Visual C++

เวอร์ชัน	รายละเอียด
MFC1.0	MFC เวอร์ชัน 1.0 ได้ถือกำเนิดขึ้นในปี ค.ศ. 1992 เป็นไลบรารีที่เพิ่มเข้าไปใน Microsoft C++ เวอร์ชัน 7 ซึ่งได้รองรับการเขียนโปรแกรมบนวินโดวส์ และ OLE 1.0
MFC2.0	ถือกำเนิดในปี ค.ศ. 1993 พร้อมกับโปรแกรม Visual C++ 1.0 ที่รู้จักกันดีว่าเป็นคอมไพเลอร์บนวินโดวส์ตัวแรก ซึ่งได้เพิ่มความสามารถในการจัดการกับโครงสร้างในระดับที่สูง และซับซ้อนได้ทั้งพร้อมรองรับกับการพัฒนาโปรแกรม 32 บิต บน Windows NT อีกด้วย
MFC2.5	ได้พัฒนาจาก MFC 2.0 สำเร็จปลายปี 1993 พร้อมกับโปรแกรม Visual C++ 1.5 ซึ่งได้เพิ่มความสามารถในการจัดการฐานข้อมูล ODBC และ OLE 2.0
MFC3.0	MFC3.0 ได้พัฒนาขึ้นพร้อมกับโปรแกรม Visual C++ 2.0 ในปลายปี ค.ศ. 1994 รองรับกับระบบ 32 บิต (Win32) ทั้งบนระบบปฏิบัติการ Windows 95, Windows NT และ Win32s รองรับการเขียนโปรแกรมโดยใช้ Thread และ Custom Control
MFC3.2	Visual 2.2 และ MFC 3.2 ได้รองรับการเขียนโปรแกรมบนระบบปฏิบัติการ Windows 95 รองรับคอนโทรลแบบใหม่ทั้งบนระบบปฏิบัติการ Windows 95 และ NT 3.51
MFC4.0	โปรแกรม Visual C++ 4.0 ได้พัฒนาขึ้นให้รองรับกับการพัฒนาโปรแกรมบนระบบปฏิบัติการ Windows 95 และ Windows NT อย่างเต็มที่รองรับอินเทอร์เน็ตเฟสใหม่ของระบบปฏิบัติการ Windows 95
MFC4.2	MFC4.2 และ Visual C++ 4.2 ถือกำเนิดในปลายปี 1996 รองรับการเขียนโปรแกรมด้วย ActiveX และการเขียนโปรแกรม Internet Server Extension (ISAPI) และคลาส Wininet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 (ต่อ) ประวัติการพัฒนาของ MFC และ Visual C++

เวอร์ชัน	รายละเอียด
MFC4.1	พัฒนาจาก MFC4.0 พร้อมกับ Visual C++ 4.1 ได้รองรับการเขียนโปรแกรมอินเทอร์เน็ตและเทคโนโลยีใหม่ๆ เช่น ISAPI เป็นต้น
MFC4.21	MFC เวอร์ชันนี้ มาพร้อมกับโปรแกรม Visual C++ 5.0 อยู่ในชุดของ Visual Studio 97 มีความสามารถครบครันในการพัฒนาโปรแกรม
MFC3.1	ต้นปี 1995 MFC3.1 และ Visual C++ 2.1 ก็ได้พัฒนาขึ้นให้รองรับการ Win 32 อย่างเต็มที่ และได้รับการเขียนโปรแกรมในระบบเน็ตเวิร์ก เช่น Windows Socket, MAPI และการควบคุมคอนโทรลใน Win32
MFC6	MFC6 และ โปรแกรม Visual C++ 6.0 เป็นโปรแกรมในชุด Visual Studio 98 ซึ่งเป็นเวอร์ชันล่าสุดที่ทำงานบนระบบปฏิบัติการ Windows 95, 98 และ NT รองรับกับ Internet Explores 4.0 และเพิ่มคอนโทรลตัวใหม่เข้าไปอีกหลายตัวเช่น คอนโทรล IP Address

ข้อดีของการเขียนโปรแกรมด้วย Visual C++ และ MFC อีกข้อหนึ่งก็คือ ความสามารถในการพอร์ตเทเบิล (Portability) หมายความว่า หากมีซอร์สโค้ดที่เขียนด้วยโปรแกรม Visual C++ ในเวอร์ชันที่ต่ำกว่า สามารถนำมาคอมไพล์และลิงค์ใหม่ได้ โดยใช้ โปรแกรม Visual C++ 6.0 หรือที่สูงกว่านี้ได้

การเขียนโปรแกรมบนวินโดวส์ด้วยโปรแกรม Visual C++ ให้ดีและมีประสิทธิภาพ จำเป็นอย่างยิ่งที่ต้องศึกษารายละเอียดและหลักการของ MFC ให้เข้าใจ อาจจะไม่ต้องศึกษาให้ลึกจนถึงระดับซอร์สโค้ดหรือการทำงานของ MFC เพียงแต่ศึกษาว่าการเขียนโปรแกรมด้วย MFC นั้นทำอย่างไร และ MFC มีประโยชน์อย่างไร

2.9.4 โปรแกรม Visual C++ 6.0 Edition

โปรแกรม Visual C++ 6.0 ได้สร้างขึ้นมา 3 รุ่น ด้วยกัน คือ Standard Edition, Professional Edition และ Enterprise Edition แต่ละรุ่นต่างก็มีความสามารถต่างกัน ดังนี้

- 1) Standard Edition ประกอบไปด้วยส่วนประกอบที่ใช้ในการสร้างและคอมไพล์โค้ด แต่จะ

ไม่มี Code Optimizations, Profiler และไม่มี Static Library ของ MFC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) Professional Edition สำหรับการพัฒนาแอปพลิเคชันระดับมืออาชีพ, การสร้าง Service และคอนโทรลสำหรับแพลตฟอร์ม 32 บิตบน 95 และ NT
- 3) Enterprise Edition สำหรับสร้างและพัฒนาแอปพลิเคชันในระดับสูงรวมทั้งระดับ Client-Server, และการเขียนโปรแกรมบน Internet และยังมีเครื่องมือใช้จัดการฐานข้อมูล SQL เช่น SQL Debugging, SQL Editing และ Visual Database Tools สำหรับออกแบบงานทางด้านฐานข้อมูล

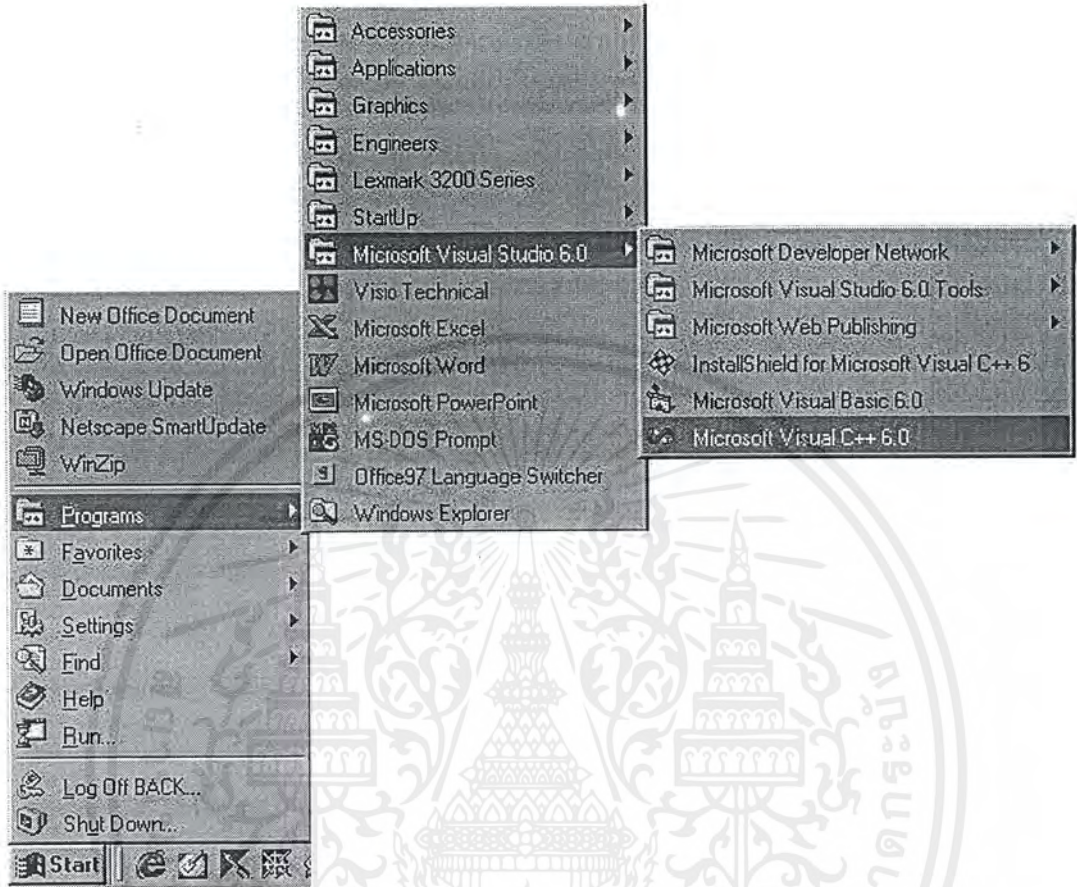
โปรแกรม Visual C++ 6.0 มีรูปแบบการอินเตอร์เฟซคล้ายกับ Visual C++ 5.0 เวอร์ชันก่อน แต่จะมีจุดหนึ่งที่แตกต่างกันคือ ระบบช่วยเหลือของ โปรแกรม Visual C++ 6.0 นั้นจะแยกออกจากตัวโปรแกรม (IDE) ซึ่งระบบช่วยเหลือจะอยู่ในรูปของ HTML Help และในการที่ใช้โปรแกรม Visual C++ 6.0 ได้นั้นจะต้องติดตั้งโปรแกรม Internet Explorer 4.0 (IE4) ลงไปก่อนด้วย หากในระบบของไม่มี IE4 โปรแกรมติดตั้ง Visual C++ จะทำการติดตั้งให้ก่อน

การที่ โปรแกรม Visual C++ 6.0 ต้องการให้ติดตั้ง IE4 ลงไปนั้นก็เพราะ โปรแกรมจะใช้ความสามารถของ IE4 ในการแสดงระบบช่วยเหลือ ซึ่งอยู่ในรูปของ HTML Help แต่ถ้าไม่ต้องการติดตั้ง IE4 ให้ติดตั้งระบบปฏิบัติการ Windows 98 ลงไปก็ได้ เพราะถ้าติดตั้งระบบปฏิบัติการ Windows 98 แล้วก็สามารถ ติดตั้งโปรแกรม Visual C++ 6.0 ลงไปได้เลย โดยไม่ต้องติดตั้ง IE4 อีก (เพราะ IE4 รมอยู่กับระบบปฏิบัติการ Windows 98 แต่ถ้ายังต้องการใช้ระบบปฏิบัติการ Windows 95 ก็สามารถติดตั้งได้โดยจะต้องติดตั้ง IE4 ก่อน)

2.10 การใช้งานโปรแกรม Visual C++ 6.0

2.10.1 การเริ่มเข้าโปรแกรม Visual C++ 6.0

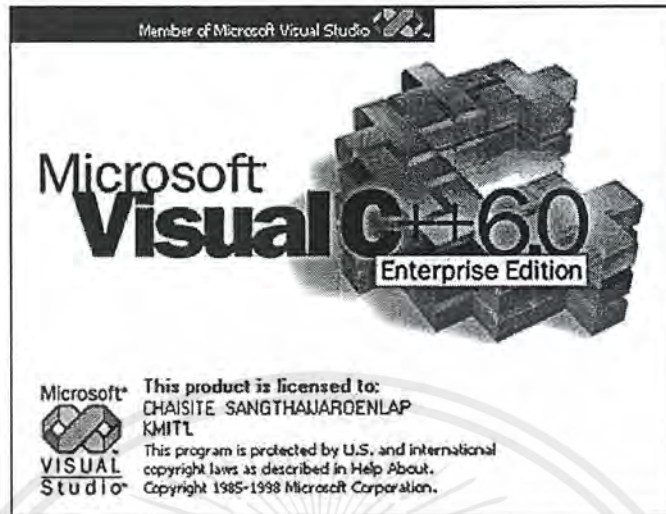
- 1) เข้าโปรแกรมวินโดวส์ปรากฏ Program manager
- 2) Click ที่ปุ่ม Start เลือกเข้าโปรแกรม Visual C++ 6.0 ดังรูปที่ 2.1



รูปที่ 2.1 การเข้าสู่โปรแกรม Visual C++ 6.0

3) ขณะที่กำลังเข้าสู่โปรแกรม Visual C++ 6.0 ปรากฏรูปดังรูปที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 โปรแกรม Visual C++ 6.0

2.10.2 การออกจากโปรแกรม Visual C++ 6.0

การออกจาก โปรแกรม Visual C++ 6.0 ทำได้หลายวิธี ดังนี้

- 1) Click ที่ Control Menu Box แล้วเลือกคำสั่ง Close
- 2) Click ที่เมนู File แล้วเลือกคำสั่ง Exit

2.10.3 โปรเจกต์เวิร์กสเปซ

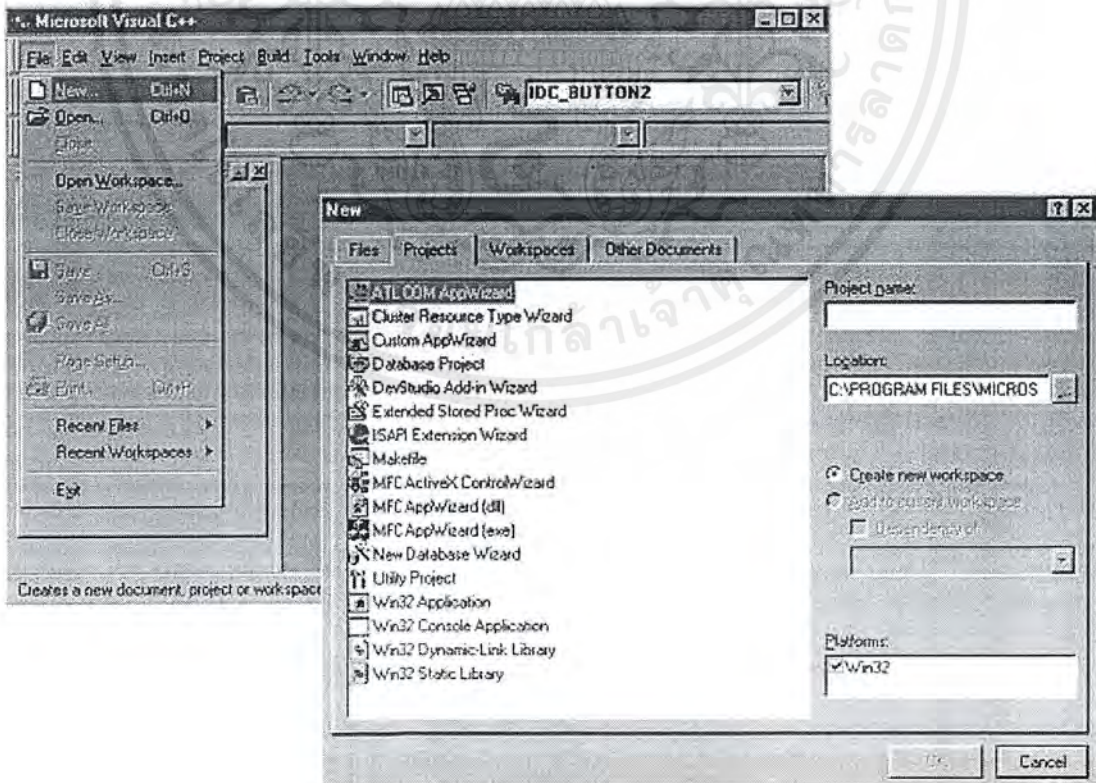
เมื่อจะทำการสร้างโปรแกรมโดยใช้ Visual C++ อันดับแรกที่ต้องทำก็คือ การสร้างโปรเจกต์เวิร์กสเปซใหม่ซึ่งเป็นการกำหนดพื้นที่ในการเก็บโปรเจกต์ที่ต้องการสร้างและกำหนดตัวเลือกต่างๆ การใช้งานโปรเจกต์ของ Visual C++ จะมีลักษณะการทำงานเหมือนกับโปรเจกต์ไฟล์ทั่วไปจะเห็นได้ว่าการเขียนโปรแกรมบนระบบดอส สามารถเขียนโปรแกรมใหญ่ๆ ได้โดยใช้ไฟล์เพียงไฟล์เดียว แต่ในการเขียนโปรแกรมบนวินโดวส์ นั้น จะต้องใช้ส่วนประกอบต่างๆ และใช้ไฟล์ร่วมกันหลายๆ ไฟล์ เพื่อให้เป็นสัดส่วนและสามารถแก้ไขโปรแกรมได้ง่ายยิ่งขึ้น โปรเจกต์ไฟล์ของ โปรแกรม Visual C++ 6.0 จะเรียกว่า “เวิร์กสเปซ” โดยจะใช้นามสกุล .DSW เป็นไฟล์ที่เก็บตัวเลือกต่างๆ ของโปรเจกต์และสามารถโหลดโปรเจกต์ที่เขียนด้วยโปรแกรม Visual C++ เวอร์ชันที่ต่ำกว่านี้ได้ เช่น .MAK หรือ .MDP เป็นต้น

ประโยชน์ของโปรเจกต์ไฟล์ สามารถสรุปได้ดังนี้

- 1) โปรเจกต์ไฟล์จะเก็บรายชื่อของไฟล์ที่เป็นซอร์สโค้ดโปรแกรมทั้งหมด ที่ใช้ร่วมกันในโปรเจกต์ เช่น ซอร์สโปรแกรม .H และ .CPP รวมทั้งไฟล์ฐานข้อมูลโปรแกรมที่ใช้ใน ClassWizard เป็นต้น
- 2) โปรเจกต์ไฟล์จะเก็บค่าตัวเลือกสำหรับการคอมไพล์ และการลิงค์เอาไว้เพื่อบอกให้รู้ว่าโปรเจกต์นี้จะทำการคอมไพล์และลิงค์กับไลบรารีใด หรือมีการสร้าง องค์ประกอบอื่นๆ อีกหรือไม่ เช่น ส่วนประกอบในการดีบั๊ก
- 3) โปรเจกต์ไฟล์จะเก็บค่าตัวเลือกที่แสดงว่าโปรเจกต์นี้เป็นโปรเจกต์แบบใดเมื่อทำการคอมไพล์ เช่น Windows Application (.EXE) หรือ Dynamic-Link Library (.DLL)

2.10.4 การจัดการโปรเจกต์เวิร์กสเปซ

ในการเริ่มต้นเขียน โปรแกรมด้วย Visual C++ ทุกครั้งจะต้องทำการสร้างโปรเจกต์เวิร์กสเปซ (Project Workspace) ขึ้นมาก่อน โดยสามารถเลือกได้ว่าจะสร้างโปรเจกต์แบบใด ซึ่งมีให้เลือกหลายแบบ ในการสร้างโปรเจกต์เวิร์กสเปซสำหรับโปรแกรมใหม่จะต้องเลือก File>New เมื่อใดจะเลือก New ปรากฏ ให้เลือกไปที่แท็บ Projects ดังรูปที่ 2.3



รูปที่ 2.3 การสร้างโปรเจกต์เวิร์กสเปซ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

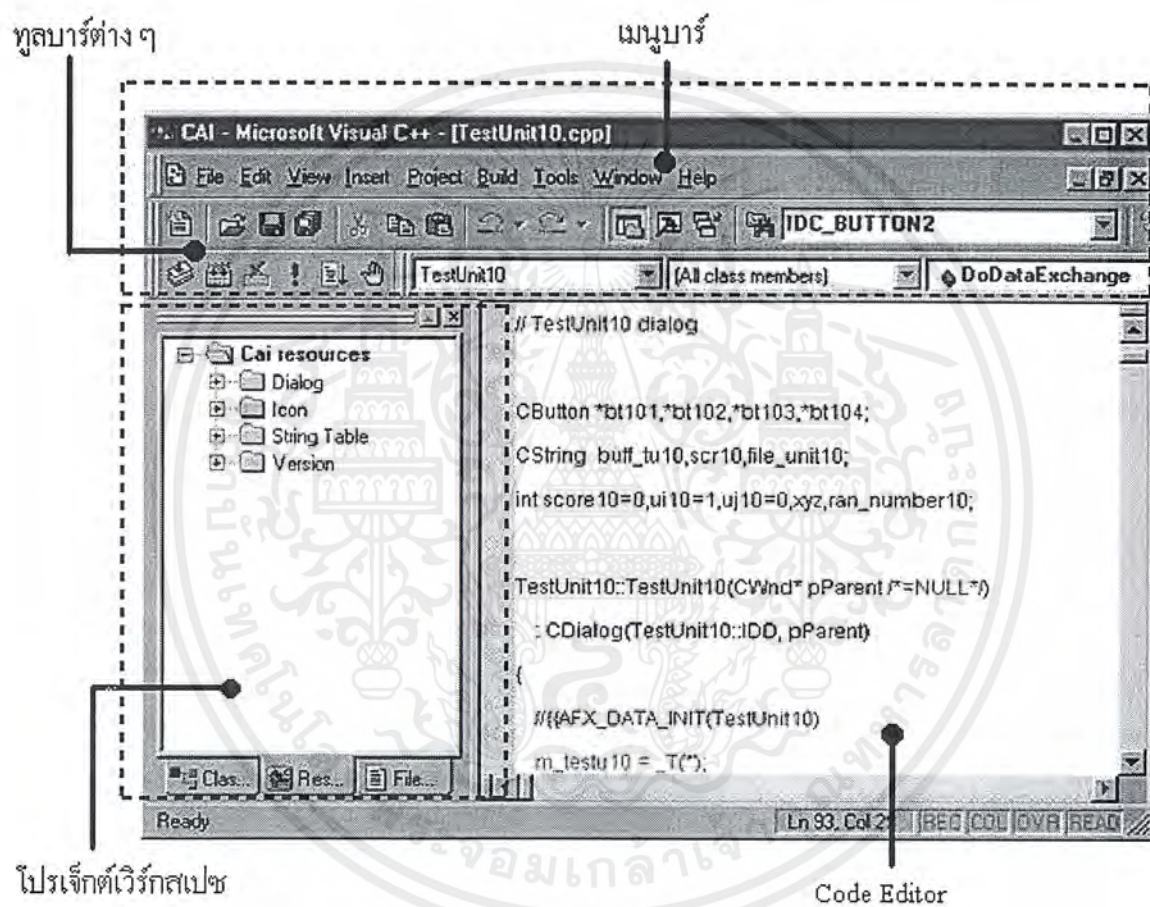
จากไดอะล็อก New จะพบกับตัวเลือกของโปรเจกต์แบบต่างๆ ซึ่งสามารถเลือกได้ว่า จะสร้างโปรเจกต์แบบใด ซึ่งจะมีอยู่ด้วยกัน 14 ตัวเลือก ดังนี้

- 1) ATL COM AppWizard เป็นการสร้างโปรเจกต์ ActiveX Control โดยใช้ ActiveX TemplateLibrary (ATL)
- 2) Custom AppWizard เป็นการสร้างตัวเลือกของAppWizard ตัวใหม่ที่สามารถกำหนดค่าต่างๆ ของโปรเจกต์ให้เหมาะกับงานที่ต้องได้
- 3) Database Project สำหรับส่วนประกอบของ DataBase ใช้สำหรับเพิ่มเข้าไปในโปรเจกต์
- 4) DevStudio Add-in Wizard เป็นการสร้างส่วนประกอบใหม่ ที่ใช้ร่วมกับ Visual Studio
- 5) ISPAI Extension Wizard เป็นการสร้าง โปรแกรม ISAPI ซึ่งเป็น DLL ที่ทำงานเหมือนกับ CGI ตัวหนึ่งใช้สำหรับสื่อสารและจัดการข้อมูลระหว่าง Sever และ Client
- 6) Makefile เป็นการสร้างโปรเจกต์ไฟล์แบบ Make File ที่มีความยืดหยุ่นในการกำหนดค่าของโปรเจกต์ด้วยตนเอง
- 7) MFC ActiveX เป็นการสร้าง ActiveX Control โดยใช้ไลบรารี MFC โดยจะมีขั้นตอนการสร้าง ControlWizard เป็นลำดับขั้นๆ ในการสร้าง
- 8) MFC AppWizard (DLL) เป็นการสร้าง Dynamic Link Library (dll) โดยใช้ไลบรารี MFC
- 9) MFC AppWizard (EXE) เป็นการสร้างโปรแกรมบนวินโดวส์ (.EXE) โดยใช้ไลบรารี MFC
- 10) New Database Wizard เป็นการสร้าง DataBase SQL Server และจัดการข้อมูลต่างๆ ทางด้านDataBase
- 11) Win32 Application เป็นการสร้างโปรแกรมบนวินโดวส์ (.EXE) แต่จะต้องเขียนโค้ดด้วยตนเอง
- 12) Win32 Console Application เป็นการสร้าง โปรแกรมใน โหมดคอนโซล (คอสโหมด)
- 13) Win32 Dynamic-Link Library เป็นการสร้าง Dynamic-Link Library (.DLL)
- 14) Win32 Static Library เป็นการสร้าง Static Library(.LIB)

การสร้างโปรเจกต์เวิร์กสเปซแต่ละแบบจะมีลักษณะที่ต่างหากต้องการเขียนโปรแกรมบนวินโดวส์ จะต้องเลือกตัวเลือก MFC AppWizard หรือ Win32 Application หรือถ้าหากต้องการสร้างโปรแกรมในโหมดคอส (Console) จะต้องเลือกไปที่ Win32 Console Application เป็นต้น

การเขียนโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0 นี้ จะใช้ MFC AppWizard ในการสร้างโปรเจกต์เป็นหลักเพราะสามารถสร้าง และแก้ไขโปรเจกต์ได้เร็วกว่าและง่ายต่อการเข้าใจ

เมื่อได้สร้างโปรเจกต์เวิร์กสเปซใหม่แล้ว โดยจะพบว่าภายในโปรเจกต์เวิร์กสเปซประกอบไปด้วยแท็บซ้อนๆ กัน 3 แท็บด้วยกันคือ ClassView, Rescuers View และ File View ซึ่งแสดงดังรูปที่ 2.4 โดยปกติแล้วถ้ายังไม่มีการสร้าง และแก้ไขโปรเจกต์ใดๆ จะไม่มีแท็บใดปรากฏเลย



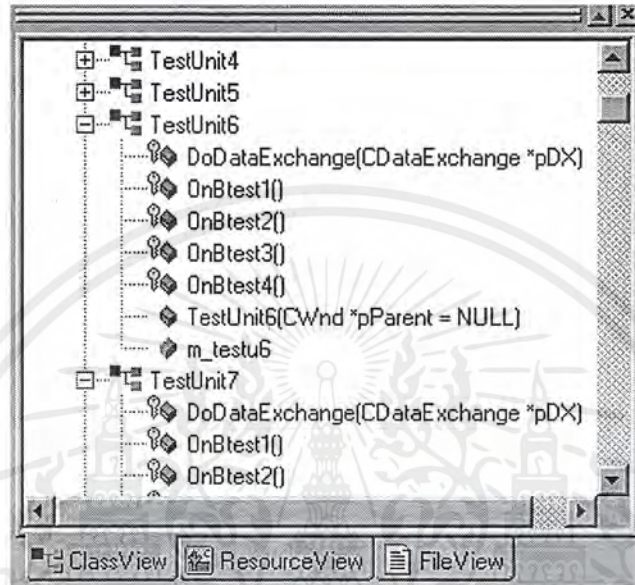
รูปที่ 2.4 โปรเจกต์ใหม่

2.10.5 รายละเอียดของแท็บต่างๆ ของโปรเจกต์เวิร์กสเปซ มีดังนี้

- 1) Class View จะใช้สำหรับแสดงรายละเอียดภายในของคลาสต่างๆที่อยู่ในโปรเจกต์นี้ สามารถรายละเอียดภายในของคลาสต่างๆที่อยู่ในโปรเจกต์นี้ สามารถทราบได้ว่ามีคลาสอะไรบ้างในโปรเจกต์ และสมาชิกของคลาสแต่ละตัว เป็นแบบ Protect หรือแบบ Public และตัวแปร โกลบอลมีอะไรบ้าง โดยจะใช้สัญลักษณ์ต่อไปนี้ ในการบอกระดับของสมาชิก

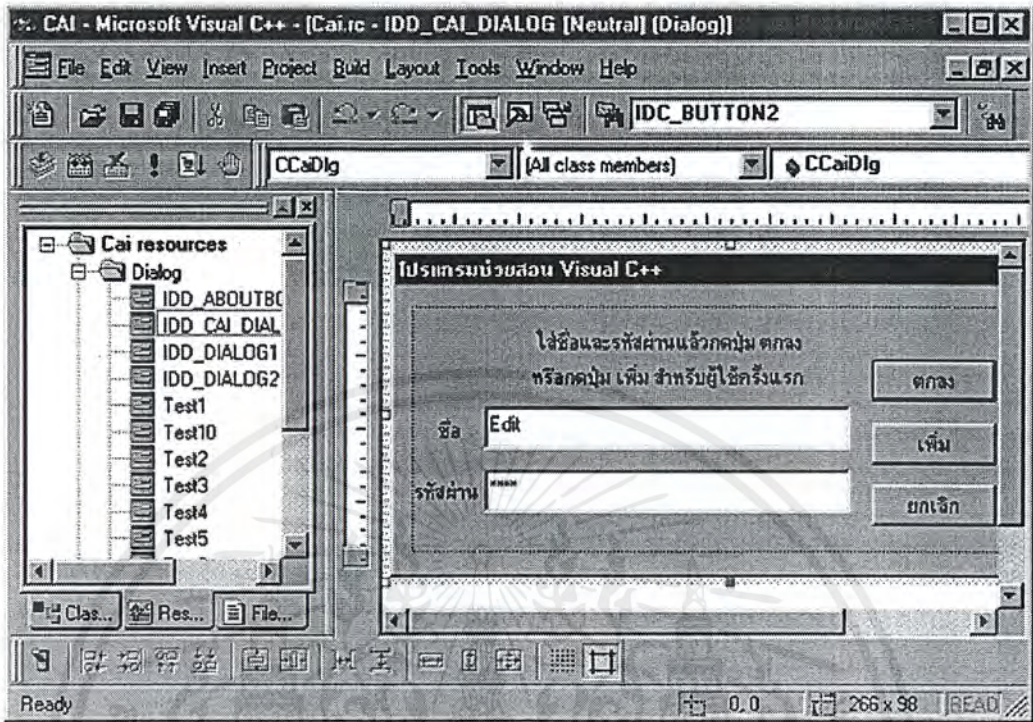
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.1 รูปแม่กุญแจ หมายถึง สมาชิกคลาสที่เป็นแบบ Private
- 1.2 รูปดอกกุญแจ หมายถึง สมาชิกคลาสที่เป็นแบบ Protect
- 1.3 รูปกล่อง หมายถึง สมาชิกคลาสเป็นแบบ Public

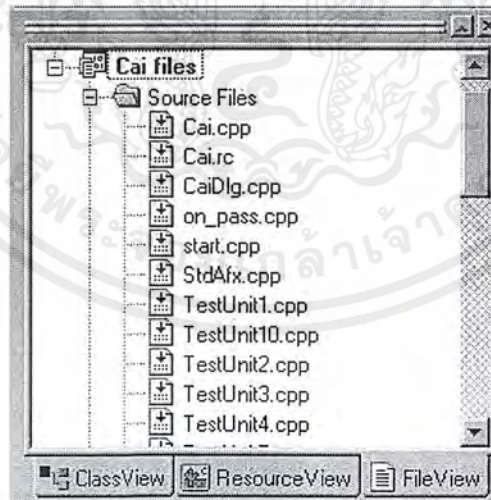


รูปที่ 2.5 Class View

- 2) **Resource View** ใช้สำหรับแสดงถึงรีซอร์สที่มีในโปรเจกต์ เช่น รูปไอคอนของโปรแกรม, ไอคอลล็อก, ทูลบาร์ และ เมนู ซึ่งสิ่งเหล่านี้สามารถสร้างขึ้นโดยใช้รีซอร์สอีดิเตอร์
- 3) **FileView** ใช้สำหรับแสดงรายชื่อไฟล์ที่อยู่ในโปรเจกต์ขณะนั้น โดย FileView จะแยกไฟล์ส่วนหัว (.H), ไฟล์โปรแกรม (.CPP) และไฟล์รีซอร์สX (.ICQ,.RC2) เอาไว้ต่างหากด้วย



รูปที่ 2.6 Resource View



รูปที่ 2.7 File View

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการแก้ไขโปรแกรม สามารถบันทึกข้อมูลลงดิสก์โดยเลือกคำสั่ง File>Save ซึ่งจะเป็นการบันทึกไฟล์ที่แก้ไขอยู่ หรือสามารถเลือกคำสั่ง File>Save Workspace เพื่อบันทึกการแก้ไขทุกอย่างในโปรเจกต์

สำหรับการเปิดโปรเจกต์เก่าขึ้นมาแก้ไข ทำได้โดยเรียกคำสั่ง Open Workspaces โดยจะพบกับไดอะล็อก Open Workspace ให้เปิดไฟล์ที่มีนามสกุล .DSW โปรเจกต์ก็จะถูกโหลดขึ้นมา และสามารถแก้ไขต่อไปได้

2.10.6 คอมไพเลอร์และลิงค์เกอร์

สิ่งหนึ่งที่จะต้องทำความรู้จักก็คือ คอมไพเลอร์ (Compiler) และลิงค์เกอร์ (Linker) เมื่อสร้างซอร์สโค้ดโปรแกรมขึ้นมา จะต้องมีการแปลภาษาในการพัฒนาโปรแกรมภาษา C ก่อนที่จะนำโปรแกรมไปใช้ได้นั้น จะต้องผ่าน 2 ขั้นตอนใหญ่ ๆ คือ การคอมไพล์ (Compile) และการลิงค์ (Link)

- 1) การคอมไพล์ (Compile) เป็นการตรวจสอบข้อผิดพลาดในซอร์สโค้ดโปรแกรมซึ่งจะใช้คอมไพเลอร์เป็นตัวจัดการหากซอร์สโค้ด โปรแกรม มีข้อผิดพลาดคอมไพเลอร์จะแสดงข้อความผิดพลาดนั้นออกมา และให้แก้ไขให้ถูกต้องก่อนนำมาคอมไพล์ใหม่ ผลที่ได้จากการคอมไพล์นั้นจะอยู่ในรูปของ ออบเจกต์ไฟล์ (Object File)
- 2) การลิงค์ (Link) การลิงค์เป็นขั้นตอนสุดท้าย โดยลิงค์เกอร์ (Linker) จะเป็นโปรแกรมที่ทำหน้าที่รวมออบเจกต์ไฟล์ที่ได้ จากการ คอมไพล์ กับฟังก์ชันจากไลบรารีภาษา C มาเป็นไฟล์ที่พร้อมนำไปใช้งาน ที่เรียกว่าไฟล์ Execute

เมื่อซอร์สโค้ดโปรแกรมผ่านกระบวนการคอมไพล์และลิงค์เรียบร้อยแล้ว ก็จะได้ไฟล์ Execute หรือไฟล์ที่สามารถนำไปใช้งานได้บนคอมพิวเตอร์วินโดวส์นั่นเอง

การพัฒนาโปรแกรมบนวินโดวส์ นั้นจะต่างจากการพัฒนาโปรแกรมในระบบดอส เช่น การคอมไพล์ซอร์สโค้ด และการคอมไพล์รีซอร์ส เพราะฉะนั้นเครื่องมือหลักๆ ที่ใช้ในกระบวนการ Build ของ โปรแกรม Visual C++ 6.0 มีดังนี้

- 1) C/C++ Compiler เป็นคอมไพเลอร์ภาษา C/C++ ของโปรแกรม Visual C++ 6.0 ซึ่งจะเป็นโปรแกรมตรวจสอบข้อผิดพลาดของซอร์สโค้ด
- 2) Resource Compiler เป็นตัวแปลภาษาของรีซอร์สสคริปต์ (Resource Script) เพราะในการพัฒนาโปรแกรมบนวินโดวส์ จะต้องมีการสร้างรีซอร์ส เช่น ไอคอน, รูปภาพ หรือ ไดอะล็อก รีซอร์สคอมไพเลอร์ตัวนี้จะเป็นโปรแกรมที่ตรวจเช็คความถูกต้องของรีซอร์ส ผลที่ได้จากการคอมไพล์ก็คือ ออบเจกต์ไฟล์ของรีซอร์ส (.RES)

- 3) Link ลิงค์เกอร์ของโปรแกรม Visual C++ 6.0 จะเป็นโปรแกรมที่ทำหน้าที่รวบรวม ออบเจกต์ไฟล์ที่ได้จากการคอมไพล์, ออบเจกต์ของรีซอร์สไฟล์ และไลบรารีไฟล์ มาสร้างเป็นไฟล์ Execute ในที่สุด

ขั้นตอนการ Build โปรแกรมของโปรแกรม Visual C++ 6.0 นั้น จะเริ่มต้นที่การคอมไพล์รีซอร์สเสียก่อน จากนั้นคอมไพเลอร์จะทำการแปลภาษา (ตรวจสอบซอร์สโค้ด) และสร้างออบเจกต์ไฟล์ขึ้นมา(สร้างไฟล์ .OBJ) และลิงค์เกอร์ก็จะทำการเชื่อมโยงไฟล์เหล่านั้นกับไลบรารีให้เป็นไฟล์ Execute (สร้างไฟล์ .EXE)

2.10.7 ทูลบาร์และเมนูบาร์ของโปรแกรม Visual C++ 6.0

ทูลบาร์ของโปรแกรม Visual C++ 6.0 มีมากมายและหลายหน้าที่การใช้งาน ซึ่งครอบคลุมการพัฒนาโปรแกรม และกระบวนการต่างๆ อย่างพร้อมสรรพทูลบาร์หลักๆ ของโปรแกรมจะมีอยู่ 8 ทูลบาร์ด้วยกัน ดังนี้

- 1) **Standard** เป็นทูลบาร์ที่ประกอบด้วยคำสั่งพื้นฐานเช่น New, Open, Save, Copy, Paste, Undo และ Redo เช่นเดียวกับทูลบาร์ของโปรแกรมไมโครซอฟต์เวิร์ด



รูปที่ 2.8 Toolbar Standard

- 2) **Build** เป็นทูลบาร์ที่ใช้ในการกำหนดตัวเลือกต่างๆของ โปรเจกต์และคอมไพล์โปรเจกต์



รูปที่ 2.9 Toolbar Build

- 3) **Build MiniBar** เป็นทูลบาร์ที่ใช้สำหรับ Build แต่จะเล็กกว่าและมีส่วนประกอบที่ใช้สำหรับ Build และการ Execute โปรแกรมโดยเฉพาะ



รูปที่ 2.10 Toolbar Build MiniBar

- 4) ATL เป็นทูลบาร์สำหรับ ActiveX Template Library (ATL) โดยจะต้องสร้างโปรเจกต์ให้รองรับกับ ATL นี้ จึงจะสามารถใช้งาน ATL Object Wizard จากทูลบาร์นี้ได้
- 5) Resource เป็นทูลบาร์ของรีซอร์สสามารถสร้างรีซอร์สต่างๆ ได้อย่างรวดเร็วโดยกดปุ่มต่างๆ ภายในทูลบาร์นี้เพื่อทำการสร้างรีซอร์ส



รูปที่ 2.11 Toolbar Resource

- 6) Edit เป็นทูลบาร์ที่ใช้สำหรับเพิ่มความสามารถในการแก้ไขซอร์สโค้ดเช่น การเพิ่มนิคมาร์ค ค้นหาที่อยู่ใน ซอร์สโค้ด



รูปที่ 2.12 Toolbar Edit

- 7) Debug เป็นทูลบาร์ที่ใช้ในการดีบักโปรแกรม เช่น Restart, Step Into และ Step Over



รูปที่ 2.13 Toolbar Debug

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 8) **DataBase** ใช้สำหรับเพิ่มและแก้ไขส่วนประกอบ ของโปรเจกต์ที่เกี่ยวข้องกับฐานข้อมูล โดยโปรแกรมที่สร้างนั้นจะต้องมีการติดต่อกับส่วนประกอบเหล่านี้ด้วย



รูปที่ 2.14 Toolbar DataBase

ในการแสดงทูลบาร์ และการยกเลิกการใช้สามารถทำได้ 2 วิธี คือ

- 1) คลิก เม้าส์ ปุ่มขวาที่ทูลบาร์เพื่อเรียก ป๊อปอัพ เมนู และกาเครื่องหมายที่หน้าชื่อทูลบาร์ที่ต้องการ
- 2) เลือกคำสั่ง Customize จากเมนู Tools เพื่อแก้ไขทูลบาร์

เมนูบาร์ (Menu Bar) ของ Visual C++ มีลักษณะพิเศษอย่างหนึ่ง ก็คือเป็นเมนูที่สามารถ ถอดออกมาจากวินโดวส์ได้ สามารถทำความเข้าใจกับเมนูต่างๆ ได้ไม่ยาก เพราะมีลักษณะเหมือนกับเมนูของโปรแกรมต่างๆ ไปรายละเอียดของเมนูแต่ละตัวมีดังนี้

- 1) เมนู **File** เป็นเมนูพื้นฐาน ที่สามารถพบได้ในเกือบทุก ๆ โปรแกรมใช้สำหรับสร้างซอร์สโค้ดใหม่หรือเปิดโปรเจกต์ขึ้นมาแก้ไข
- 2) เมนู **Edit** เป็นเมนูที่ใช้ในการแก้ไขซอร์สโค้ด เช่น Copy, Paste
- 3) เมนู **View** ใช้สำหรับแสดงหน้าต่างและเปลี่ยนมุมมองการแก้ไข โปรเจกต์ เช่นการ แสดงหน้าต่าง Properties หน้าต่าง Output หรือการปรับให้อยู่ในโหมด Full Screen รวมทั้งการเรียกโปรแกรม Class Wizard
- 4) เมนู **Insert** ใช้สำหรับการเพิ่มคลาสใหม่ และรีซอร์สหรือส่วนประกอบของ AT (Active Template Library)
- 5) เมนู **Project** ใช้สำหรับปรับแต่งตัวเลือกต่างๆ ของโปรเจกต์ เช่น การเพิ่มซอร์สไฟล์เข้าไปในโปรเจกต์ และการกำหนดตัวเลือกการคอมไพล์
- 6) เมนู **Build** เป็นเมนูที่ใช้ในการคอมไพล์โปรเจกต์และการรันโปรแกรม
- 7) เมนู **Tools** เป็นเมนูที่ใช้สำหรับเรียกใช้เครื่องมือต่างๆ และใช้ในการปรับแต่งโปรแกรม
- 8) เมนู **Window** เป็นเมนูที่ใช้สำหรับจัดเรียงวินโดวส์ภายในโปรแกรม เช่น Cascadeเรียงวินโดวส์ในแนวนอนและแนวตั้ง และปิดวินโดวส์ทั้งหมด เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 9) เมนู Help ใช้สำหรับแสดงระบบช่วยเหลือและรายละเอียดของผลิตภัณฑ์ต่างๆรวมทั้งการกำหนดบู๊ทมารีคให้กับ Info Viewer และยังสามารถเปิดโฮมเพจของบริษัทไมโครซอฟท์ได้จาก Info Viewer โดยตรงอีกด้วย

2.10.8 รายละเอียดและการใช้งาน Help

Help หรือระบบช่วยเหลือของโปรแกรม Visual C++ 6.0 จะอยู่ในรูปแบบเอกสาร HTML โดยใช้ส่วนประกอบของ Inter Explorer 4.0 ในการแสดงผล HTML ระบบช่วยเหลือของโปรแกรม Visual C++ 6.0 จะอยู่แยกจากตัวโปรแกรม Visual C++ 6.0 ต่างจากระบบช่วยเหลือในเวอร์ชัน 5.0 ระบบช่วยเหลือของโปรแกรม Visual C++ 6.0 นี้เรียกว่า MSDN Library ซึ่งจะเป็นระบบช่วยเหลือแบบ HTML Help แทนการใช้ไฟล์ช่วยเหลือแบบเก่า (. HLP) จึงเป็นเหตุผลที่ว่าทำไมเวลาติดตั้งโปรแกรม Visual C++ 6.0 จึงต้องติดตั้งโปรแกรม Internet Explore 4 เสียก่อน

MSDN Library จะแบ่งออกเป็น 2 ส่วนใหญ่ๆ

- 1) ส่วนที่แสดงสารบัญและ Index ต่างๆ จะอยู่ทางด้านซ้ายของหน้าต่าง MSDN Library ซึ่งจะมีอยู่ 4 แท็บ คือ Content, Index, Search และ Favorites
- 2) ส่วนแสดงเอกสาร จะอยู่ทางด้านขวาของหน้าต่าง MSDN Library มีลักษณะการแสดงผลเป็น HTML เหมือนกับการใช้โปรแกรม Internet Expole 4.0 แสดงโฮมเพจๆ หนึ่ง ซึ่งสามารถคลิกที่ตัวอักษรที่ขีดเส้นใต้เพื่อลิงค์ไปยังเอกสารที่เกี่ยวข้องได้

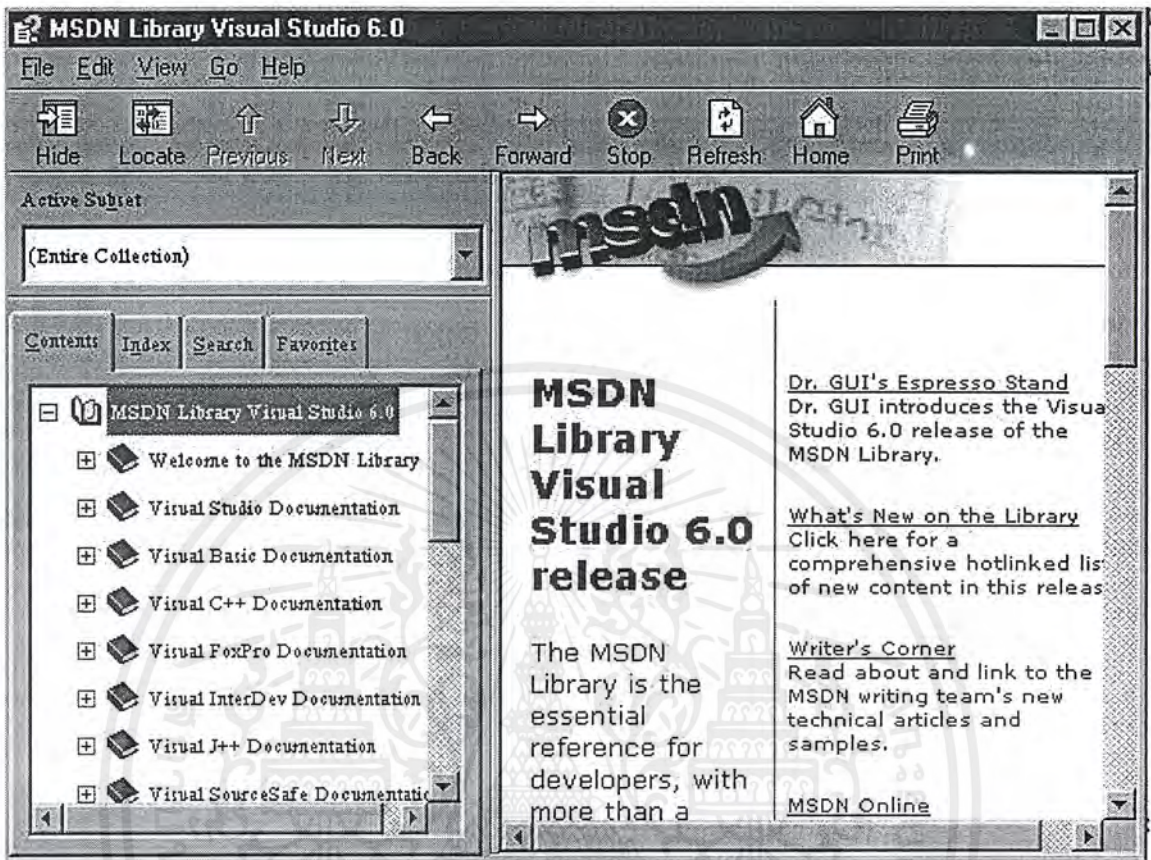
จากซีกซ้ายของ MSDN จะประกอบด้วยแท็บ 4 แท็บ ดังนี้

- 1) Content เป็นสารบัญของระบบช่วยเหลือ
- 2) Index เป็นดัชนีคำที่ต้องการค้นหา
- 3) Search เป็นการค้นหาคำหรือคีย์เวิร์ดที่ต้องการ
- 4) Favorites จะแสดงคำหรือคีย์เวิร์ดที่ค้นหาไปแล้ว หรือที่ค้นหาย่อย

การค้นหาเอกสารโดยใช้ Index สามารถทำได้โดยพิมพ์ข้อความหรือหัวข้อที่ต้องการลงในช่อง “Type in the keybord find” เมื่อพิมพ์ลงไปรายการในลิสต์ด้านล่างก็จะเลื่อนไปยังข้อความนั้นๆ ให้โดยอัตโนมัติ ให้ ดับเบิล คลิก ไปยังหัวข้อที่ต้องการ หรือกดปุ่ม Display

การค้นหาแบบ Search จะมีความละเอียดมากกว่าแบบ Index เพราะจะเป็นการค้นหาคำหรือประโยคที่อยู่ภายในเอกสาร การค้นหาทำได้โดยใส่ข้อความที่ต้องการลงในช่อง “Type in the word(s) to search for” จากนั้นกดปุ่ม List Topic หรือ Enter เพื่อทำการค้นหาโปรแกรมจะทำการค้นหาและแสดงรายการของหัวข้อที่พบคำหรือข้อความดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 หน้าต่างของ MSDN Library

ในการแสดงเอกสารและคู่มืออ้างอิงในการเขียนโปรแกรม เช่น รายละเอียดของฟังก์ชัน main() หรือรายละเอียดของคลาสต่างๆ สามารถทำได้โดยการกดปุ่ม F1 เมื่อกำลังแก้ไข โปรแกรมอยู่ หน้าต่าง Help ก็จะมาปรากฏและแสดงรายละเอียดของฟังก์ชันตัวนั้นออกมาให้

2.10.9 การสร้างไอคอน

อุปกรณ์ที่ใช้พัฒนาริซอร์สที่มีประสิทธิภาพอีกตัวหนึ่งไม่แพ้เมนูอิดิตเตอร์นั่นก็คือ ไอคอนอิดิตเตอร์ โดยสามารถกำหนดและแก้ไขตำแหน่งการวางของคอนโทรลและเซ็คค่า ID ของคอนโทรลแต่ละตัวได้ง่ายๆ โดยการคลิกเมาส์ลากและวาง เหมือนกับการพัฒนาโปรแกรมด้วยโปรแกรม Visual Basic หรือโปรแกรม Delphi ไอคอนอิดิตเตอร์ก็มีลักษณะเช่นเดียวกัน

การเพิ่มการสร้างไอคอนเข้าไปในโปรเจกต์ Icon นี้โดยให้ปฏิบัติตามขั้นตอนต่อไปนี้

1) ขั้นตอนที่ 1 สร้างไอคอนใหม่

ให้เลือกเมนู Insert>Resource จากนั้นดับเบิ้ลคลิกไปที่ Dialog ก็จะพบกับไอคอนอิดิต

เตอร์พร้อมชุดคอนโทรลหลายอย่าง จะเห็นได้ว่าชุดทูลบาร์ที่ใช้สร้างไอคอน จะมีปุ่มให้เลือกเอกสารเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำเมาส์ไปคลิกที่ปุ่มใดก็ตาม ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากมายหลายปุ่ม หากเคยใช้ โปรแกรม Visual Basic หรือ โปรแกรม Delphi มาก่อน ก็จะสามารถเข้าใจกับปุ่มเหล่านี้ได้ไม่ยาก หากต้องการจะดูว่าในเวลารันโปรแกรม ใดอะลือกจะมีหน้าต่างอย่างไร ไม่ต้องไปเสียเวลากับการคอมไพล์และรันเพื่อดูผลการทำงานแต่ให้กดปุ่ม <Ctrl+T> หรือปุ่มที่มีรูปร่างคล้ายกับสวิทช์ไฟในทูลบาร์ด้านล่าง (ปุ่มแรกทางด้านซ้ายมือสุด) ซึ่งเป็นปุ่มทดสอบ ใดอะลือกเพื่อให้สามารถเห็นหน้าต่างของใดอะลือกได้ทันที



รูปที่ 2.16 ทูลบาร์สำหรับสร้างใดอะลือก

2) ขั้นตอนที่ 2 ปรับแต่งใดอะลือก

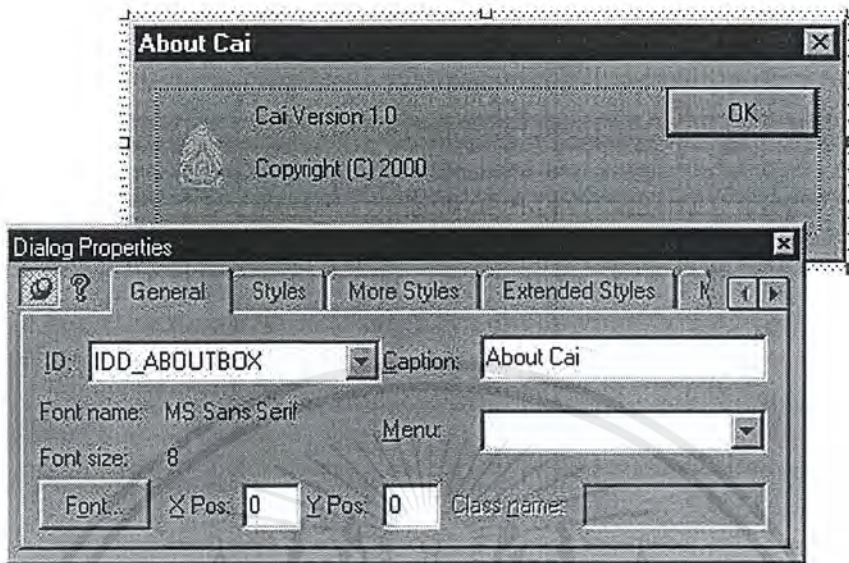
เมื่อสร้างใดอะลือกขึ้นใหม่ ภายในใดอะลือกใหม่นั้นจะเป็นใดอะลือกเปล่าๆและจะมีปุ่มกดมาให้ 2 ปุ่มด้วยกัน นั่นก็คือ ปุ่ม OK และปุ่ม Cancel ให้ลบปุ่ม Cancel ออกไปโดยใช้เมาส์ Click ที่ปุ่ม และกดปุ่ม Del จากคีย์บอร์ด และจัดปุ่ม OK ให้อยู่ในตำแหน่งตรงกลางด้านล่างของใดอะลือก

3) ขั้นตอนที่ 3 เปลี่ยนค่า Resource ID ของใดอะลือก

ให้ดับเบิลคลิก ที่พื้นที่ใดอะลือกตรงจุดใดก็ได้ที่ไม่ใช่อุปกรณ์หรือคอนโทรล จะพบใดอะลือก Properties ปรากฏขึ้นมา จะเห็นได้ว่าใดอะลือกอดีตเตอร์ได้เซ็ค่า Resource ID ของใดอะลือกให้มีค่าเริ่มต้นเป็น IDD_DIALOG1 ให้ทำการเปลี่ยนค่าให้เป็นค่าใหม่ เพราะจะทำให้ไม่สับสนในการเขียนโปรแกรมจากเดิม IDD_DIALOG1 ให้เปลี่ยนเป็น IDD_ABOTIDIALIG จากนั้นเปลี่ยน Caption ให้เป็น About Program

จากนั้น ให้ปรับแต่งใดอะลือกตามต้องการเพราะใดอะลือก About นี้จะใส่อะไรที่เกี่ยวกับ

ผลิตภัณฑ์หรือโปรแกรมลงไป เช่น ช่องโปรแกรม และ ปีที่สร้างและที่อยู่ติดต่อเมื่อต้องการ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 การแก้ไขค่า ID และ Caption

4) ขั้นตอนที่ 4 เพิ่มโค้ดโปรแกรมให้เรียกไดอะล็อก

ไดอะล็อกที่สร้างขึ้นใหม่ จะยังไม่สามารถใช้งานได้จนกว่าจะสร้างคลาสขึ้นมา ควบคุมมัน จะต้องเขียนโค้ดโปรแกรมเพิ่มเติมลงใน Icon.cpp โดยจะสร้างคลาสใหม่ที่ชื่อ CAboutDlg โดยให้สืบทอดมาจากคลาส CDialog

2.10.10 การสร้าง Hotkey

จากตัวอย่างที่ผ่านนี้จะเห็นได้ว่าสามารถเข้าถึงข้อมูลถึงเมนูเหล่านั้นได้โดยใช้ เมาส์ คลิกไปที่เมนูหรือใช้คีย์บอร์ดเลือกเมนูดังกล่าว ยังมีวิธีหนึ่งที่สามารถจะทำให้เข้าถึงเมนูนั้นๆ ได้อย่างรวดเร็วกว่าใช้ เมาส์ หรือคีย์บอร์ด นั่นคือการใช้ฮ็อตคีย์ (Hot Keys)

ฮ็อตคีย์หรือคีย์ลัด เป็นคีย์ที่สามารถกดเพื่อให้เข้าถึงเมนูนั้นได้อย่างรวดเร็ว เช่นในโปรแกรม NotePad ถ้าต้องการก๊อปปี้ข้อมูลลงคลิปบอร์ดก็สามารถกด <Ctrl+C> ได้ทันที หรือหากต้องการค้นหาที่กดแค่ปุ่ม F3 ไดอะล็อก Search ก็จะปรากฏ วิธีการนี้จะทำให้สามารถใช้งานได้ อย่างรวดเร็วขึ้น มากกว่าการใช้ เมาส์ อย่างเดียว

โปรแกรมตัวอย่างในบทนี้ที่ได้ทำการสร้างไปนั้น สามารถควบคุมการกดปุ่มฮ็อตคีย์ ได้โดยใช้รีซอร์สสร้าง Accelator (ตารางฮ็อตคีย์) ขึ้นมาจากนั้นก็เรียกฟังก์ชัน

LoadAccelTable() เพื่อให้ฮ็อตคีย์ทำงาน

ขั้นตอนการสร้างตารางฮ็อตคีย์นั้นไม่มีอะไรมาก อันดับแรกให้เลือกเมนู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Insert>Resource และให้เลือกไปที่ Accelerator เมื่อคลิกปุ่ม New แล้วก็จะพบกับหน้าต่างแบบตาราง ดังรูปที่ 2.18



รูปที่ 2.18 การเพิ่ม ฮ็อตคีย์ ให้ไอคอน

สามารถเพิ่ม Hot Key ได้โดยการ ดับเบิลคลิก ที่บรรทัดว่าง เพื่อเรียกไอคอน Properties จากนั้นจึงกำหนดค่า ID และค่า Hotkey ในไอคอน Properties จากนั้นให้ Save การแก้ไข Resource และเขียนโปรแกรมติดตั้ง Accelerator เพื่อให้มันสามารถทำงานได้ จะต้องเรียกฟังก์ชัน LoadAccelTable() เพื่อใส่โค้ดนี้ไว้ในฟังก์ชันคอนสตรัคเตอร์ของคลาส Cwin

2.10.11 AppWizard

เป็นอุปกรณ์ตัวหนึ่งที่ใช้สำหรับสร้างซอร์สโค้ดโปรแกรมต้นแบบ ซึ่งจะช่วยอำนวยความสะดวกให้กับเป็นอย่างมากเพราะไม่จำเป็นต้องเขียนโค้ดโปรแกรม ด้วยตนเองการใช้งาน AppWizard ก็ทำได้ง่ายๆ คือ AppWizard จะถามเกี่ยวกับรูปแบบของโปรแกรมตัวเลือกที่ต้องการ จากนั้น AppWizard ก็จะสร้างซอร์สโค้ดโปรแกรมขึ้นมาให้โดยอัตโนมัติ สามารถนำซอร์สโค้ดนี้ไปใช้งานได้ทันที ซึ่งจะช่วยอำนวยความสะดวกให้กับผู้เขียนโปรแกรมเป็นอย่างมาก

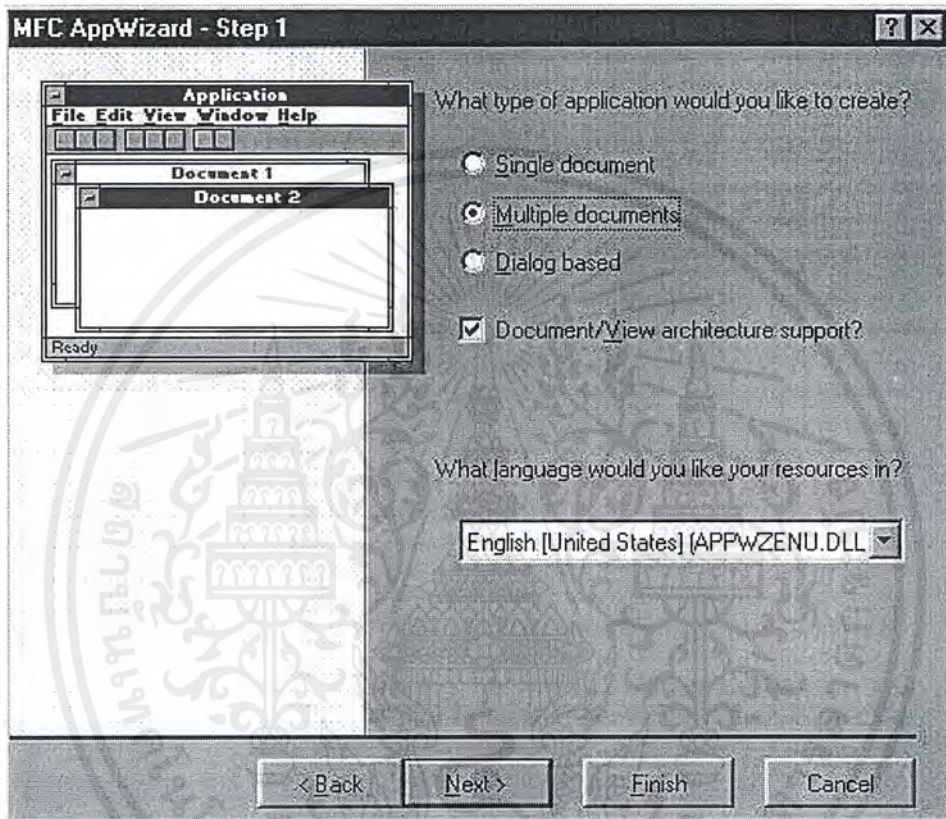
ตัวเลือกที่จะต้องกำหนดให้ AppWizard ตัวหนึ่งก็คือ รูปแบบของโปรแกรมซึ่ง AppWizard จะให้ตัวเลือกมา 3 ตัวโดยสามารถเลือกรูปแบบของโปรแกรมที่ต้องการได้ 3 แบบ คือ

- 1) โปรแกรมแบบ SID (Single Document Interface) เป็นการสร้างโปรแกรมในลักษณะ

ของ Notepad

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) โปรแกรมแบบ MID (Multiple Document Interface) เป็นการสร้างโปรแกรมที่มีหลายหน้าต่างคล้ายๆ กับ Microsoft Word
- 3) โปรแกรมแบบ Dialog-Based เป็นการสร้างโปรแกรมแบบไดอะล็อก



รูปที่ 2.19 MFC AppWizard

ในการเลือกรูปแบบของโปรแกรม จะต้องพิจารณากันสักเล็กน้อย เพราะหากเลือกรูปแบบของโปรแกรมผิดแล้ว การเปลี่ยนแปลงและการเพิ่มเติมจะทำได้ค่อนข้างยาก

เมื่อเลือกรูปแบบของโปรแกรมที่ต้องการสร้าง และกำหนดรายละเอียดให้กับ AppWizard เรียบร้อยแล้ว AppWizard ก็จะทำการสร้างซอร์สโค้ดต้นแบบขึ้นมาให้ สามารถคอมไพล์และรันโปรแกรมได้ทันที โปรแกรมที่ได้จะมีรูปแบบเหมือนกับที่ได้กำหนดใน AppWizard ทุกประการ

ซอร์สโค้ดที่ AppWizard สร้างขึ้น จะเป็นโค้ดโปรแกรมที่ใช้ MFC เป็นหลัก และ AppWizard จะจัดโครงสร้างของไฟล์ในโปรเจกต์ให้ง่ายต่อการทำความเข้าใจ เพราะส่วนที่เป็นการประกาศคลาส และตัวแปรต่างๆ จะถูกเก็บอยู่ในไฟล์ส่วนหัว (.H) และส่วนที่เป็นเนื้อหาของ

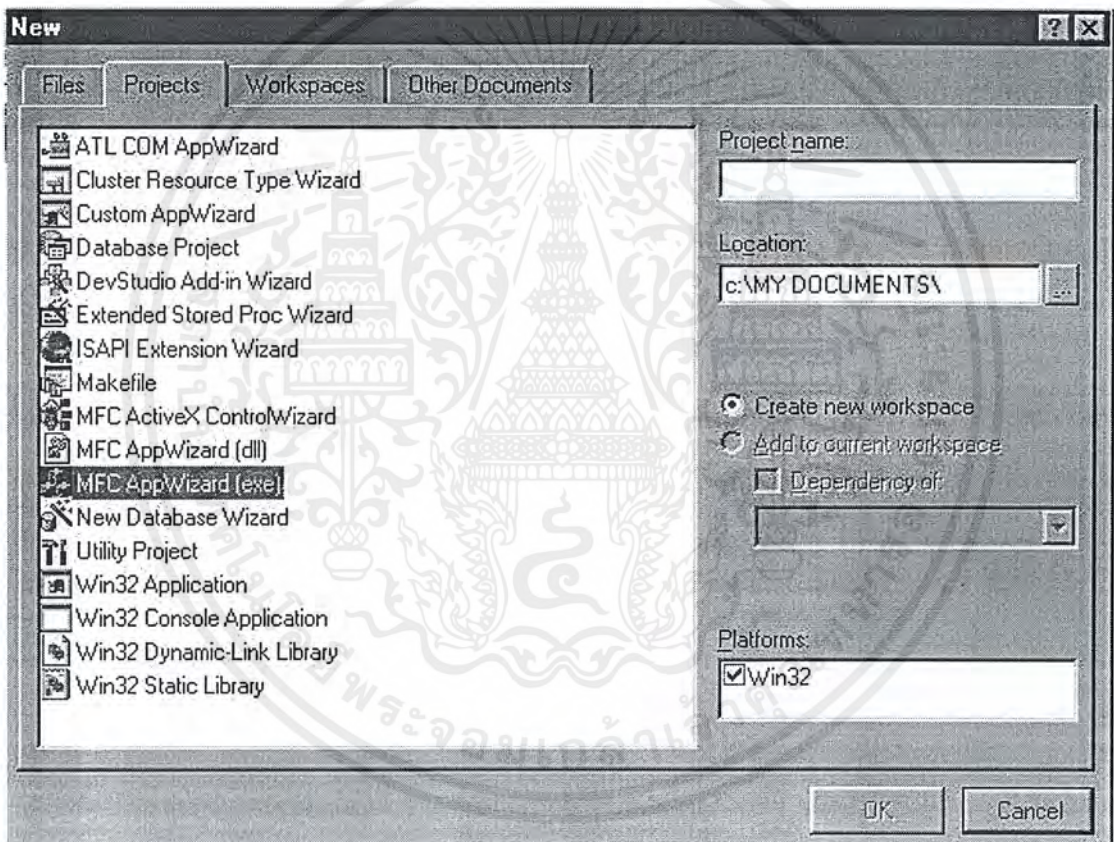
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันจะเก็บไว้ในไฟล์โปรแกรม (.CPP) โดยจะแยกกันอยู่คนละไฟล์จึงง่ายต่อการแก้ไข และดัดแปลงซอร์สโค้ด ซึ่งจะลองมาทำความรู้จักและลองใช้งาน AppWizard ด้วยกัน

2.10.12 การสร้างโปรแกรมแบบ Dialog-Based

การจะใช้ AppWizard สร้างโปรแกรมแบบ Dialog-Base ในระดับพื้นฐานขึ้นมา โดยจะมาทดลองสร้างและรันโปรแกรมไปพร้อมๆ กัน ให้ปฏิบัติตามขั้นตอนขั้นตอนการสร้างโปรเจกต์ดังต่อไปนี้

1) ขั้นตอนที่ 1 สร้างโปรเจกต์เวิร์กสเปซใหม่



รูปที่ 2.20 การสร้างโปรเจกต์เวิร์กสเปซเลือกเมนู

File>New จะพบกับไดอะล็อก New ซึ่งภายในจะมีแท็บซ้อนๆ กันทั้งหมด 4 แท็บ ให้เลือกไปที่แท็บที่ 2 คือ Project ดังรูปที่ 2.20

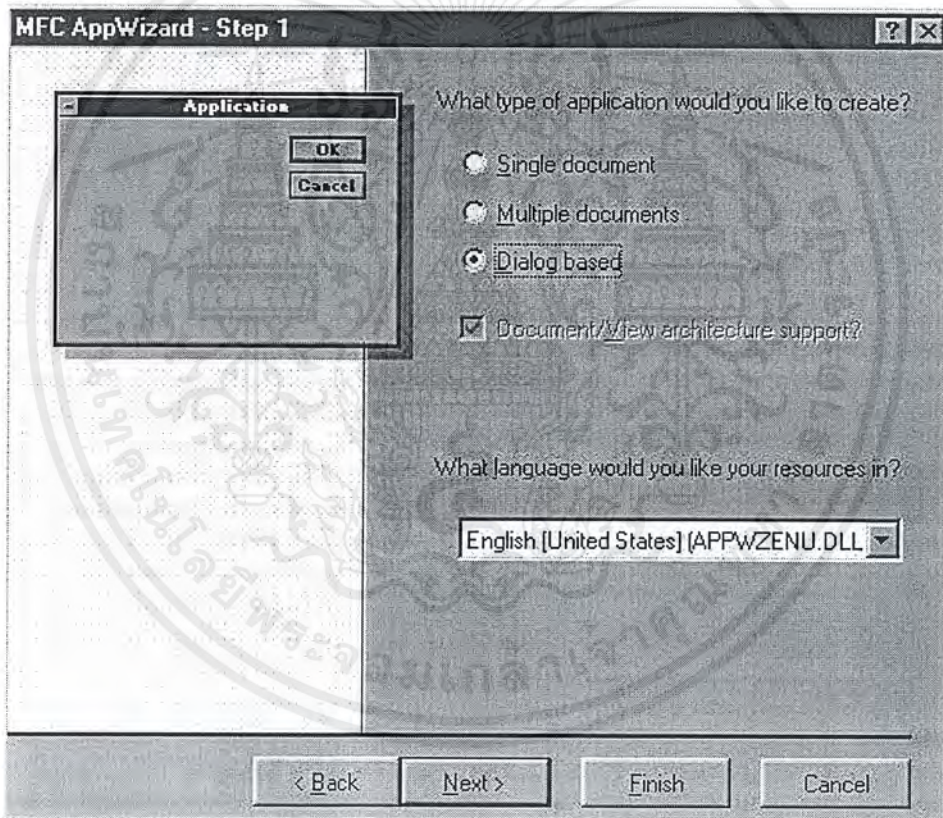
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ขั้นตอนที่ 2 เข้าเมนูการทำงานของ AppWizard

ในแท็บ Project ให้คลิก เลือก MFC AppWizard (exe) จากนั้นกำหนดชื่อโปรเจกต์ในช่อง Project Name ให้เป็น Basic และกำหนดไดร์ฟและไดเรกทอรีที่เก็บโปรเจกต์ในช่อง Location โดยสามารถคลิกปุ่มเพื่อเลือกไดเรกทอรีจากไดอะล็อก Choose Directory ได้

เมื่อกำหนดรายละเอียดของโปรเจกต์ครบหมดแล้ว ให้กดปุ่ม OK เพื่อเข้าสู่ AppWizard โดย AppWizard จะทำการสร้างไดเรกทอรีใหม่ให้กับโปรเจกต์โดยอัตโนมัติ (โดยจะใช้ชื่อของโปรเจกต์เป็นชื่อของไดเรกทอรี ถ้ากำหนดชื่อโปรเจกต์ว่า CAI ก็จะได้ชื่อไดเรกทอรีใหม่ว่า CAI)

3) ขั้นตอนที่ 3 เลือกรูปแบบของโปรแกรม



รูปที่ 2.21 การเลือกรูปแบบโปรแกรม

เมื่อไดอะล็อก MFC AppWizard Step 1 ปรากฏ ซึ่งจะให้เลือกรูปแบบของโปรแกรม โดยมี 3 ตัวเลือก คือ Single Document, Multiple Document และ Dialog Based

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตอนนี้นำโปรเจกต์ Basic ที่จะสร้างเป็นโปรเจกต์แบบง่ายๆ เพื่อทดสอบการทำงานให้เลือกไปที่ตัวเลือกที่ 3 คือ Dialog Based ส่วนตัวเลือกอื่นให้ปล่อยไว้ตามเดิม ดังรูปจากนั้นกดปุ่ม <Next> เพื่อเข้าสู่กระบวนการต่อไป

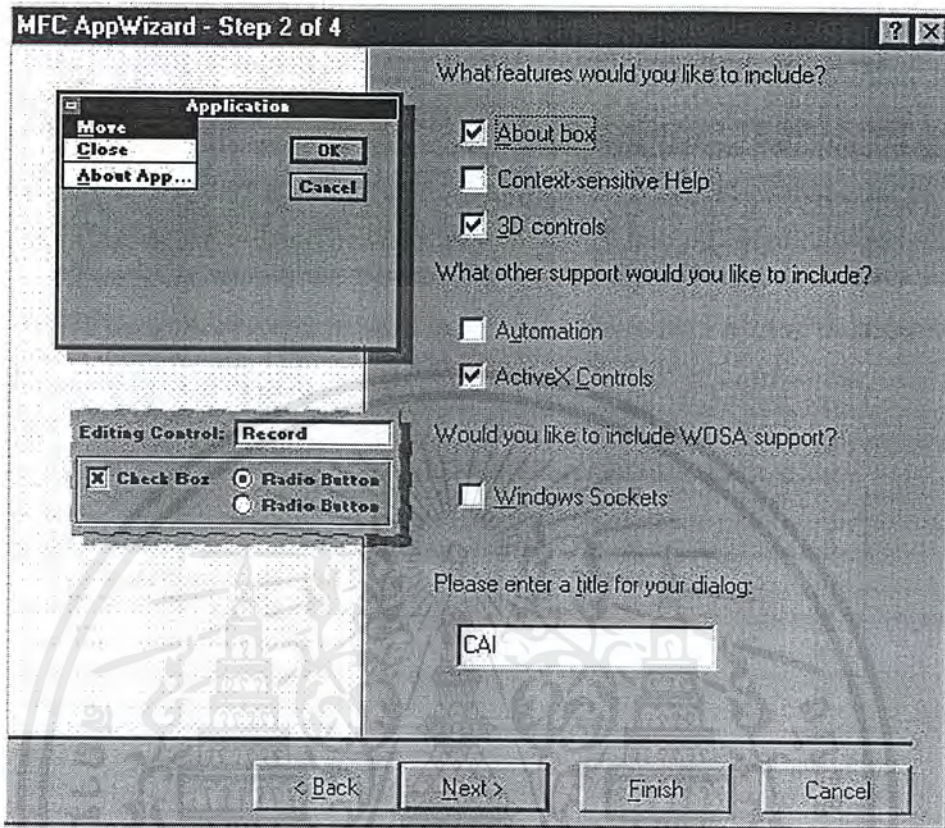
4) ขั้นตอนที่ 4 กำหนดรายละเอียดที่ต้องการให้กับโปรแกรม

จะพบกับไดอะล็อก MF AppWiZard Step 2 Of 4 ซึ่งคำว่า Step 2 Of 4 จะบอกว่าขั้นตอนการสร้างมีทั้งหมด 4 ขั้นตอนด้วยกัน และตอนนี้อยู่ในขั้นตอนที่ 2 ซึ่งก็คือการกำหนดรายละเอียดให้กับโปรเจกต์ที่ต้องการ

จากไดอะล็อก MFC App WiZard Step 2 Of 4 ให้กำหนดค่าดังนี้

- 1) ยกเลิกเครื่องหมายในช่อง About Box และ Context-sensitive Help ให้เป็นช่องว่างๆ เพราะยังไม่จำเป็นต้องใช้ About Box และไม่จำเป็นต้องสร้าง Online Help
- 2) ทำเครื่องหมายใน ช่อง 3D Control เพื่อให้การแสดงผลของวินโดวส์เป็นแบบ 3 มิติ (ดูสวยและน่าใช้กว่า)
- 3) ยกเลิกเครื่องหมายในช่อง Automation,ActiveX Controls และ Windows Sockets ออกทั้งหมด เพราะยังไม่จำเป็นต้องใช้
- 4) ภายในช่อง “Please enter a title for your dialog” ให้ใส่ข้อความอะไรก็ได้ที่ต้องการ ข้อความนี้ก็คือ ข้อความบนไตเติ้ลบาร์ของโปรแกรมนั่นเอง ถ้าไม่แก้ไขก็จะแสดงชื่อของโปรเจกต์

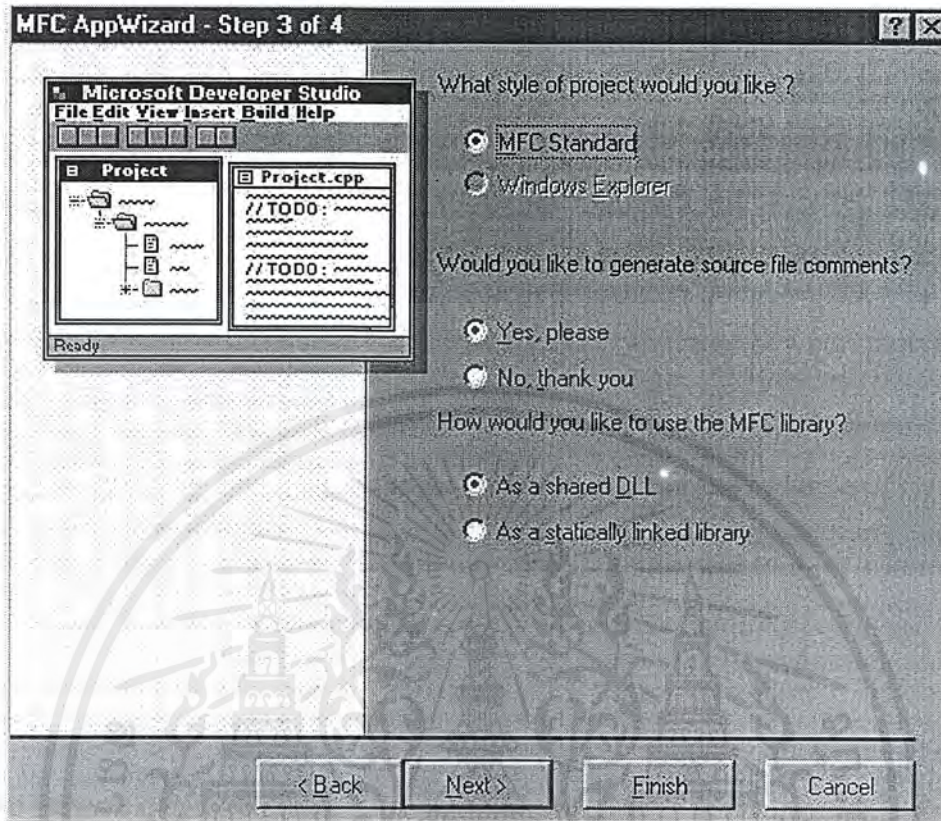
เมื่อกำหนดตัวเลือกเรียบร้อยแล้วให้กดปุ่ม Next เพื่อเข้าสู่ขั้นตอนต่อไปของAppWiZard



รูปที่ 2.22 การกำหนดรายละเอียดให้กับโปรแกรม

5) ขั้นตอนที่ 5 กำหนดตัวเลือกของโปรเจกต์

ต่อไปจะเป็นขั้นตอนที่ 5 จะพบไดอะล็อก MFC App Wizard Step 3 of 4 ซึ่งให้กำหนดตัวเลือก 2 ตัวคือ ตัวเลือกของการสร้างคำอธิบายโปรแกรมและรูปแบบการ Link ของโปรแกรม ดังรูปที่ 2.23



รูปที่ 2.23 การกำหนดตัวเลือกของโปรเจกต์

จากไดอะล็อก MF AppWizard Step 3 of 4 จะเห็นได้ว่า App Wizard จะถาม 2 คำถามด้วยกัน (จริงๆ แล้วจะถาม 3 คำถาม แต่คำถามบนสุด จะเป็นคำถามที่เกี่ยวกับการเลือกรูปแบบของโปรแกรม ซึ่งถูก Disable เอาไว้)

ในคำถามแรกจะถามว่า “ต้องการสร้างคำอธิบาย (คอมเมนต์) ให้กับซอร์สโค้ดหรือไม่” คำอธิบายนี้จะมีประโยชน์ในการทำความเข้าใจกับโปรแกรมแต่ก็จะทำให้โค้ดดูรก ในตอนนี้ให้ตอบ No, thank you

และอีกคำถามหนึ่งก็คือ “ต้องการลิงค์ (Link) กับไลบรารี MFC แบบใด” ซึ่งจะมีตัวเลือกให้เลือก 2 ตัวคือ As a Shared DLL และ As a statically linked library

ความแตกต่างของการลิงค์ทั้งสองแบบนี้ก็คือ ถ้าลิงค์แบบ Shared DLL เมื่อนำเอาโปรแกรมไปรันบนวินโดวส์ในเครื่องอื่น ในระบบนั้นจะต้องมีไฟล์ MFC42.DLL และ MSVCRT.DLL อยู่ด้วย ซึ่งถ้าไม่มีไฟล์ .DLL ทั้งสองตัวนี้โปรแกรมก็จะไม่สามารถทำงานได้ ข้อดีของการลิงค์แบบนี้คือไฟล์ .EXE ที่ได้จะมีขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ถ้าลิงค์แบบ Statically linked library เมื่อนำเอาโปรแกรมไปรันที่เครื่องอื่นไม่จำเป็นต้องนำไฟล์.DLL ไปด้วย แต่ไฟล์.EXE ที่ได้จะมีขนาดใหญ่กว่าเดิมเล็กน้อย

ในตัวอย่างแรกๆ นี้ ให้เลือกไปที่ “As a shared DLL” ก่อน เมื่อกำหนดตัวเลือกเสร็จเรียบร้อยแล้ว ดังรูป ให้กดปุ่ม Next เพื่อเข้าสู่ขั้นตอนต่อไป

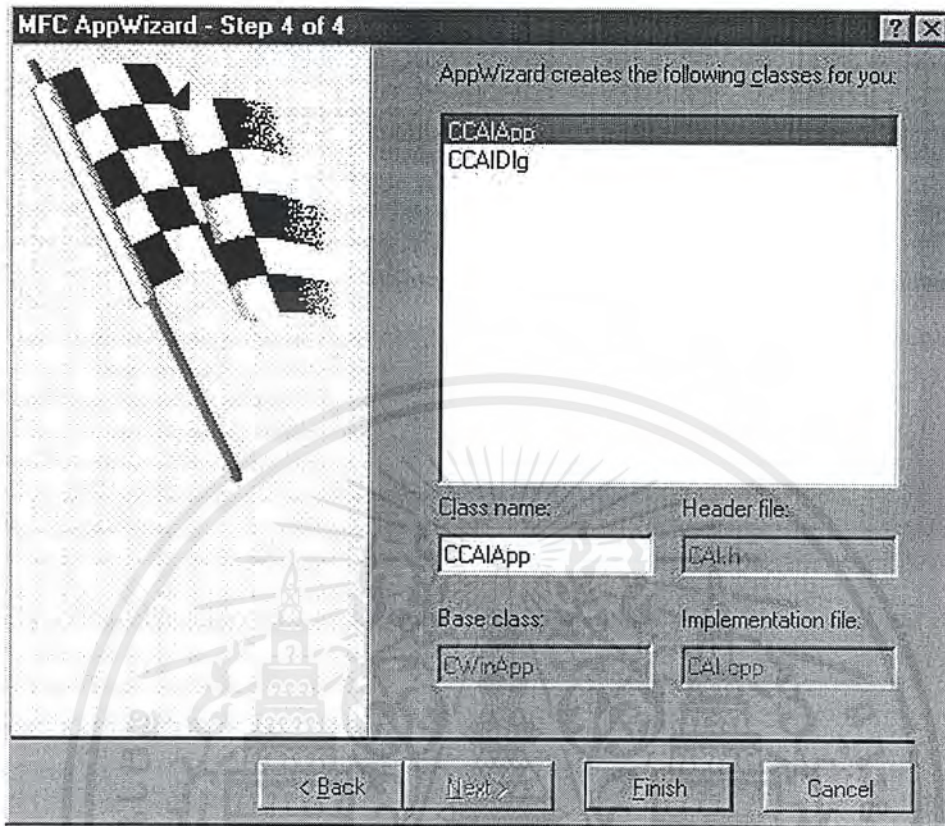
6) ขั้นตอนที่ 6 ขั้นสุดท้ายขั้นตอนการสร้างโปรเจกต์

ไอคอน MFC AppWizard Step 4 of 4 จะเป็นรูปธงแสดงถึงสิ้นสุดการสร้างโปรเจกต์ AppWizard จะแสดงรายละเอียดของของซอร์สไฟล์ที่ AppWizard จะสร้างขึ้นมาให้ ดังรูปที่ 2.24 ในตอนนี้ AppWizard จะสร้างคลาสขึ้นมาให้ 2 คลาสด้วยกัน คือ

- 1) คลาส CCAIApp คลาสนี้จะสืบทอดมาจากคลาส CWinApp ของ MFC ใช้ในการควบคุมการทำงานของโปรแกรมทั้งหมด เป็นคลาสที่สำคัญที่สุดในโปรแกรม Basic
- 2) คลาส CCAIDlg คลาสนี้จะสืบทอดมาจากคลาส Cdialog ของ MFC ทำหน้าที่ควบคุมการแสดงผลและการทำงานของอุปกรณ์ ส่วนประกอบต่างๆ ภายในไอคอน

ในตอนนี้ยังไม่ต้องสนใจรายละเอียดของซอร์สโค้ดโปรแกรมมากนัก เพราะจะมาศึกษากันอย่างละเอียดต่อไป หลังจากที่ได้ทดลองรันโปรแกรมทดสอบการทำงานกันเรียบร้อยแล้ว

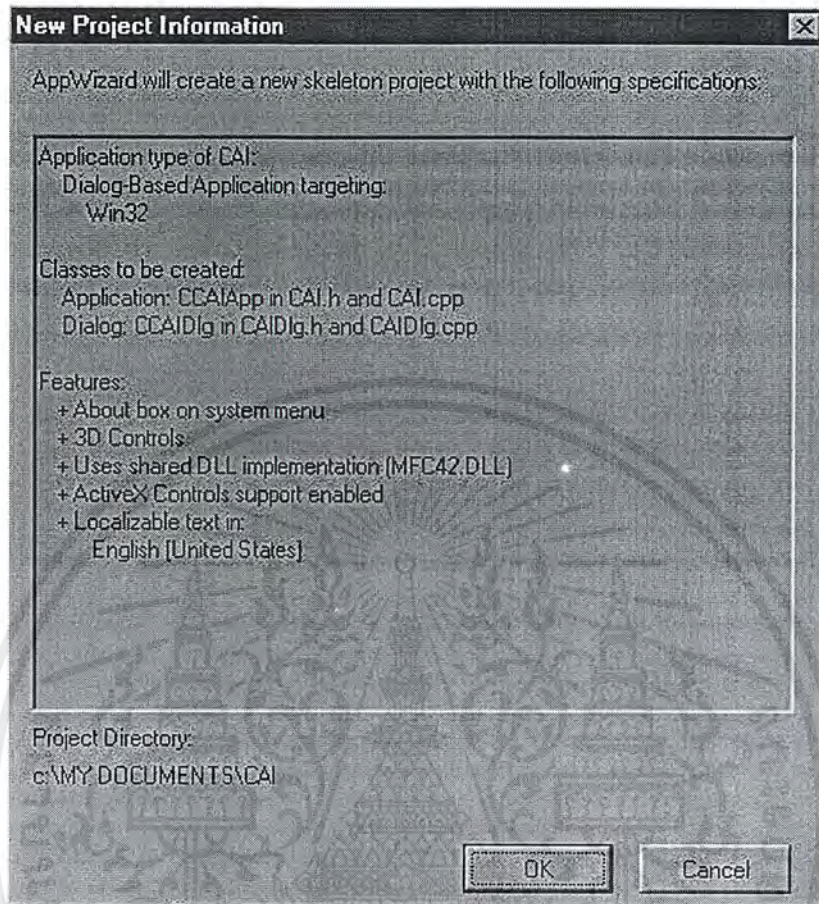
ให้กดปุ่ม Finish จะพบกับไอคอน New Project Information ดังรูปที่ 2.25 ซึ่งไอคอนนี้จะแสดงรายงานการกำหนดตัวเลือกของอีกครั้ง ให้กดปุ่ม OK เพื่อยืนยันความต้องการในการสร้างโปรเจกต์ แต่ถ้าดูจากรายงานแล้วพบว่ากำหนดตัวเลือกไม่ครบก็ให้กดปุ่ม Cancel และกด Back กลับ ไปแก้ไข



รูปที่ 2.24 การสิ้นสุดการสร้างโปรเจกต์

ในการสร้างโปรเจกต์โดยใช้ AppWiZard ควรจะตรวจเช็คตัวเลือกต่างๆ ให้รอบคอบ มิฉะนั้นถ้า AppWiZard สร้างโปรเจกต์ให้กับแล้ว ถ้าต้องการเปลี่ยนแปลงใดๆ ต้องแก้ไขโค้ดโปรแกรมเอง

เมื่อ AppWizard สร้างโปรเจกต์เวิร์กสเปซ CAI เรียบร้อยแล้ว จะพบกับโปรเจกต์เวิร์กสเปซ และไดอะล็อกอิดิตเตอร์ที่จะใช้ สำหรับสร้างและแก้ไขไดอะล็อก



รูปที่ 2.25 รายละเอียดของโปรเจกต์ที่สร้างขึ้น

2.10.13 การใช้งานไดอะล็อกอิดิตเตอร์

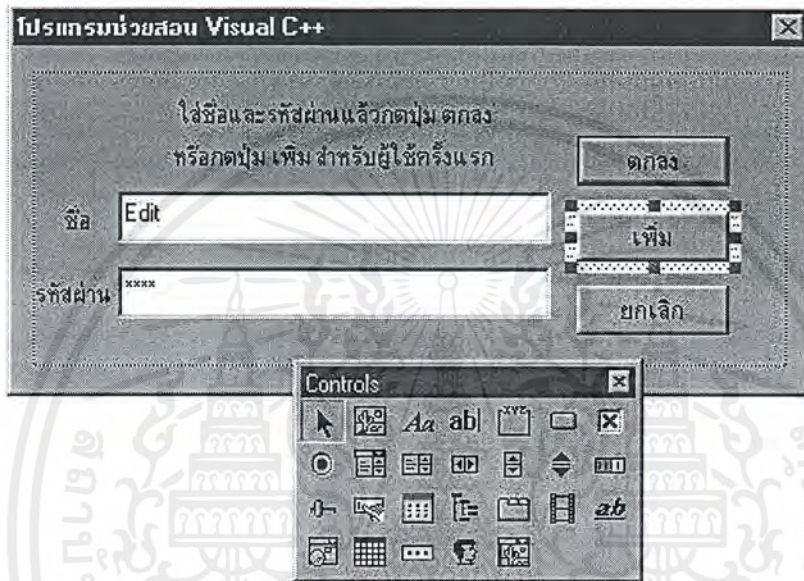
การสร้างคอนโทรลหรืออุปกรณ์ภายในไดอะล็อก โดยจะใช้ไดอะล็อกอิดิตเตอร์ในการสร้าง โดยรูปแบบของไดอะล็อกอิดิตเตอร์จะมีลักษณะเป็นแบบ Visual หรือ WYSIWYG คล้ายกับการสร้างโปรแกรมด้วยโปรแกรม Visual Basic หรือโปรแกรม Delphi แต่ไดอะล็อกอิดิตเตอร์ของโปรแกรม Visual C++ 6.0 จะใช้ในการออกแบบไดอะล็อกในโปรแกรม ซึ่งจะไม่มีการเขียนโค้ดโปรแกรมลงไปแบบในโปรแกรม Visual Basic แต่จะเขียนโค้ดเพื่อเชื่อมโยงไปยังไดอะล็อกเหล่านั้นอีกทีหนึ่ง

ไดอะล็อกทุกอันที่สร้างขึ้น โดยไดอะล็อกอิดิตเตอร์นั้นจะมีค่า ID กำกับทุกอันเพื่อใช้ในการอ้างถึงเวลาเขียนโปรแกรม ให้เปิดไดอะล็อกรีซอร์ส IDD_BASIC_DIALOG จะพบกับไดอะล็อกอิดิตเตอร์ที่แสดงไดอะล็อก IDD_BASIC_DIALOG ให้ดูจะเห็นว่ามียูลบาร์ที่เข้าร่วมกับไดอะล็อกอิดิตเตอร์ 2 ยูลบาร์ด้วยกัน คือยูลบาร์ของคอนโทรล และยูลบาร์ควบคุมการจัดวาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.14 การวางคอนโทรลในไดอะล็อก

ในการนำคอนโทรลมาวางภายในไดอะล็อก ทำได้โดยการคลิกเมาส์ไปยังคอนโทรล ที่ต้องการในทูลบาร์ จากนั้นลากมาวางลงในไดอะล็อกในตำแหน่งที่ต้องการ จากนั้นสามารถปรับขนาดให้ใหญ่เล็กได้ตามต้องการ ดังรูปที่ 2.26



รูปที่ 2.26 การวางคอนโทรลลงในไดอะล็อก

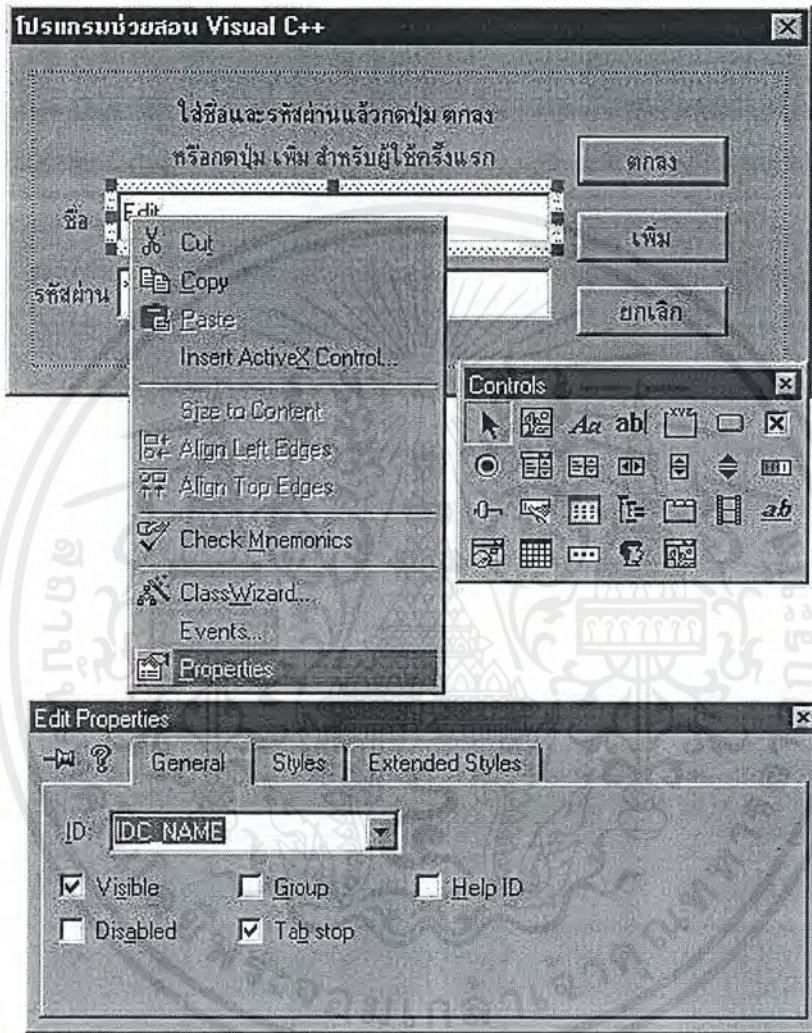
การนำคอนโทรลมาวางจะเหมือนกับการใช้โปรแกรมทั่วไปเช่น โปรแกรมวาดภาพ, Word ซึ่งสามารถศึกษาวิธีการนำคอนโทรลมาวางได้ไม่ยาก และนอกจากนี้ยังสามารถกำหนดระยะได้โดยใช้ เมาส์ และปุ่มลูกศรบนคีย์บอร์ด ได้อีกด้วย

2.10.15 การกำหนดค่าคงที่ให้กับคอนโทรล

คอนโทรลทุกตัวที่นำมาวางลงในไดอะล็อกกรีซอร์ส จะต้องมีค่าคงที่ (ID) เพื่อใช้ในการอ้างถึงตัวคอนโทรลในเวลาที่เขียนโค้ดโปรแกรม ตัวอย่าง เช่น ID_LISTBOX, ID_NAME และ ID_LASTNAME เป็นต้น เมื่อสร้างคอนโทรลตัวใหม่ลงในไดอะล็อกกรีซอร์สไดอะล็อกอิดิตเตอร์ จะกำหนดค่า ID ให้กับคอนโทรลโดยอัตโนมัติ แต่ไม่ควรใช้ค่าที่กำหนดมาในนี้ เพราะ ไดอะล็อกอิดิตเตอร์จะกำหนด ID โดยไล่ตามลำดับเป็นก่อนหลังเป็นหมายเลข เช่น ID_CONTRO1 และ D_CONTROL2, ID_CONTROL3 ไปเรื่อยๆ ซึ่งยากต่อการจดจำ ควรจะเปลี่ยนค่า ID ของคอนโทรลเหล่านี้ให้สอดคล้องกับการทำงานของคอนโทรลตัวนั้น เช่น ถ้าสร้างปุ่มกดที่ใช้สำหรับจบโปรแกรม ก็ควรกำหนดค่า ID ให้เป็น ID_EXIT เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปลี่ยนค่า ID ของคอนโทรลในไคอะล๊อค สามารถทำได้โดยการคลิกเมาส์ ปุ่มขวามบนคอนโทรลและเรียกคำสั่ง Properties จากเมนูป๊อปอัพ ดังรูปที่ 2.27



รูปที่ 2.27 การกำหนดค่าคงที่ให้กับคอนโทรล

2.10.16 การทดสอบไคอะล๊อค

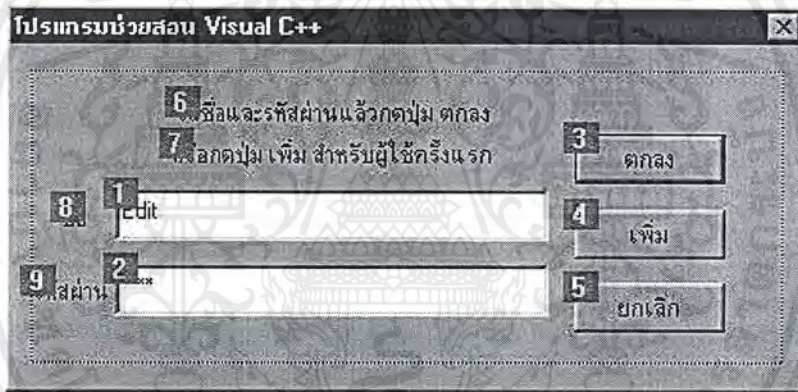
เมื่อแก้ไขไคอะล๊อคกรีซอร์ส อาจต้องการทดสอบว่า ตำแหน่งการวางไคอะล๊อค และระยะโพกัสได้พอดีหรือเปล่านั้นไม่ต้องทำการ Build เพื่อดูผลการทำงาน ซึ่งจะเสียเวลามาก สามารถทดสอบไคอะล๊อคได้โดยกดปุ่ม <Ctrl+T> หรือเลือกคำสั่ง test จากเมนู Layout หรืออาจจะกดปุ่มที่มีรูปร่างคล้ายสวิตช์ที่ทูลบาร์ Dialog ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.17 การโฟกัสของคอนโทรล

การที่คอนโทรลอยู่ในสถานะพร้อมรอรับคำสั่งจากผู้ใช้สามารถรู้ได้ว่าตอนนี้ คอนโทรลตัวใด ในไดอะล็อกที่กำลังรอรับคำสั่งหรือถูกโฟกัสอยู่ โดยดูจากตำแหน่งของเส้นประที่ล้อมรอบคอนโทรลนั้น ดังรูปแสดงให้เห็นถึงการโฟกัสของ คอนโทรลซึ่งจะอยู่ที่ปุ่ม OK เป็นปุ่มแรก

ตำแหน่งการโฟกัสของคอนโทรลสามารถย้ายไปยังคอนโทรลตัวอื่นได้ โดยการกดปุ่ม <Tab> ตำแหน่งของโฟกัสจะวนไปยังคอนโทรลทุกๆ ตัวภายในไดอะล็อกตามลำดับการสร้างคอนโทรล ซึ่งเรียกลำดับนี้ว่า แท็บออเดอร์ (Tab Order) สามารถกำหนดลำดับหรือเปลี่ยนลำดับของ Tab Order ให้โฟกัสไปยังคอนโทรลที่ต้องการเพื่อความเหมาะสมในการทำงานได้โดยเรียกคำสั่ง Layout>Tab Order หรือกด <Ctrl+D> ดังรูปที่ 2.28



รูปที่ 2.28 การกำหนดลำดับโฟกัสคอนโทรล

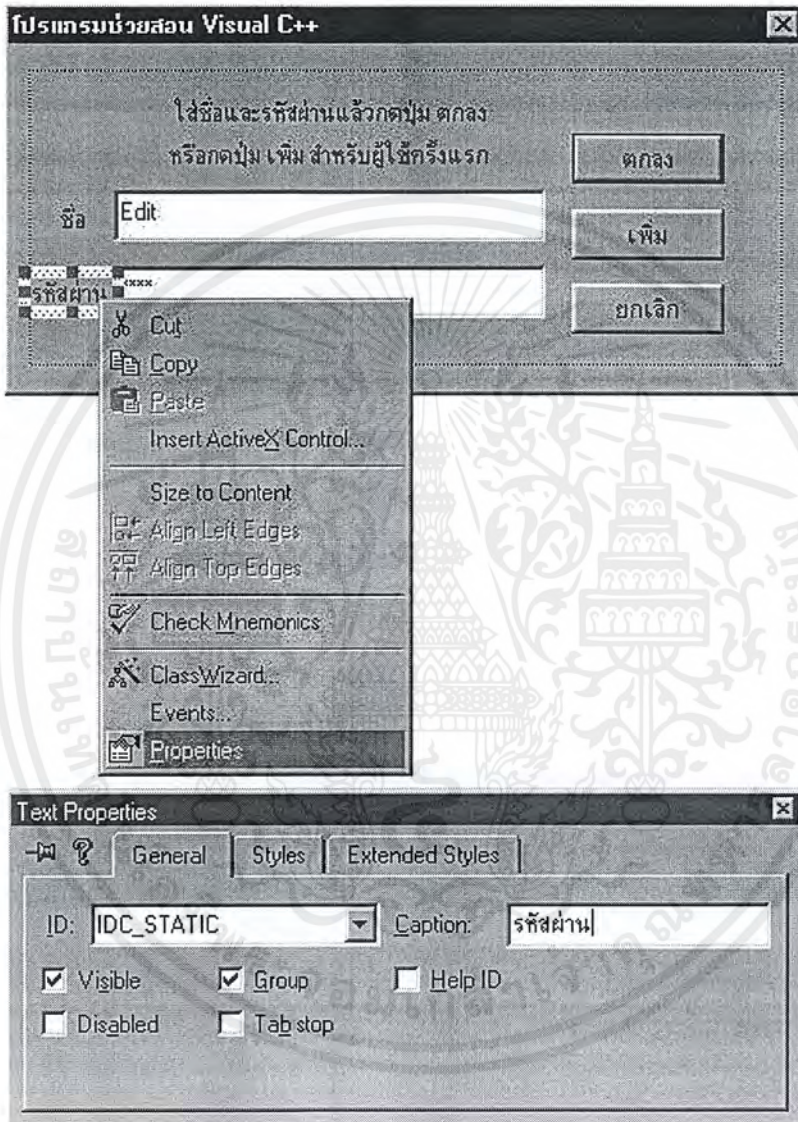
2.10.18 การแสดงข้อความในไดอะล็อก

ทำการสร้างสแตติกคอนโทรลใหม่โดยใช้การคลิกเมาส์ไปที่สแตติกคอนโทรลในคอนโทรลทูลบาร์จากนั้นนำมาวางลงในไดอะล็อกและปรับแต่งขนาดตามต้องการ

เมื่อได้ทำการเพิ่มสแตติกคอนโทรลลงในไดอะล็อกแล้วให้เลือกสแตติกคอนโทรล และคลิกเมาส์ ปุ่มขวาเพื่อเรียกเมนูป๊อปอัพ โดยให้เลือกไปที่ คำสั่ง Properties ก็จะพบกับ ไดอะล็อก Text Properties

ไดอะล็อก Text Properties ที่ปรากฏขึ้นนี้ จะใช้สำหรับกำหนดข้อความที่แสดงในสแตติกคอนโทรลและกำหนดค่า ID ของคอนโทรลนั้น รวมไปถึงรูปแบบการแสดงผลของคอนโทรลด้วย จากไดอะล็อก Text Properties ให้พิมพ์ข้อความที่ต้องการลงในช่อง Caption เช่น “รหัสผ่าน” ดังรูปที่ 2.29

ส่วนข้อความที่อยู่ภายในช่อง ID นั้นไม่จำเป็นต้องเปลี่ยนแปลง เพราะจะให้คอนโทรลสแตติกตัวนี้ แสดงข้อความเพียงอย่างเดียว มันจึงถูกกำหนดเป็น IDC_STATIC อย่างเดิม



รูปที่ 2.29 การแสดงข้อความลงในไอคอน

2.10.19 การสร้างรูปภาพภายในไอคอน

ในการแสดงรูปภาพในไอคอน สามารถใช้คอนโทรลสแตติกนี้แสดงรูปภาพได้ 2 แบบ
คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1) แสดงรูปภาพที่เป็น ไอคอน (.ICO)
- 2) แสดงรูปภาพที่เป็น ไฟล์รูปภาพ

ในการสร้างไฟล์รูปภาพและไฟล์ไอคอนนั้น สามารถใช้ Image Editor สร้างได้ หรือ Import ไฟล์รูปภาพเข้ามาก็ใช้ได้

2.10.20 การแสดงรูปภาพที่เป็นไอคอน

ให้ทำการสร้างไอคอนริชอร์ส โดยเลือกไปที่แท็บ Resource View ของโปรเจกต์ จากนั้นเลือกคำสั่ง Insert>Resource หรือกดปุ่ม <Ctrl+R> ก็จะพบกับไดอะล็อก Insert Resource

ดังรูปที่ 2.30 จากนั้นให้เลือกไปที่ Icon และกดปุ่ม New เพื่อสร้างไอคอนริชอร์ส

จากนั้นจะพบกับ ไอคอนอิดิเตอร์ (Icon Editer) ที่ใช้ในการสร้างและแก้ไขไอคอนให้กับโปรเจกต์ วาดรูปอะไรก็ได้ลงไป โดยหยิบเครื่องมือต่างๆ มาใช้จากทูลบาร์ทางด้านขวา เช่นเดียวกับการใช้โปรแกรมวาดภาพทั่วไป

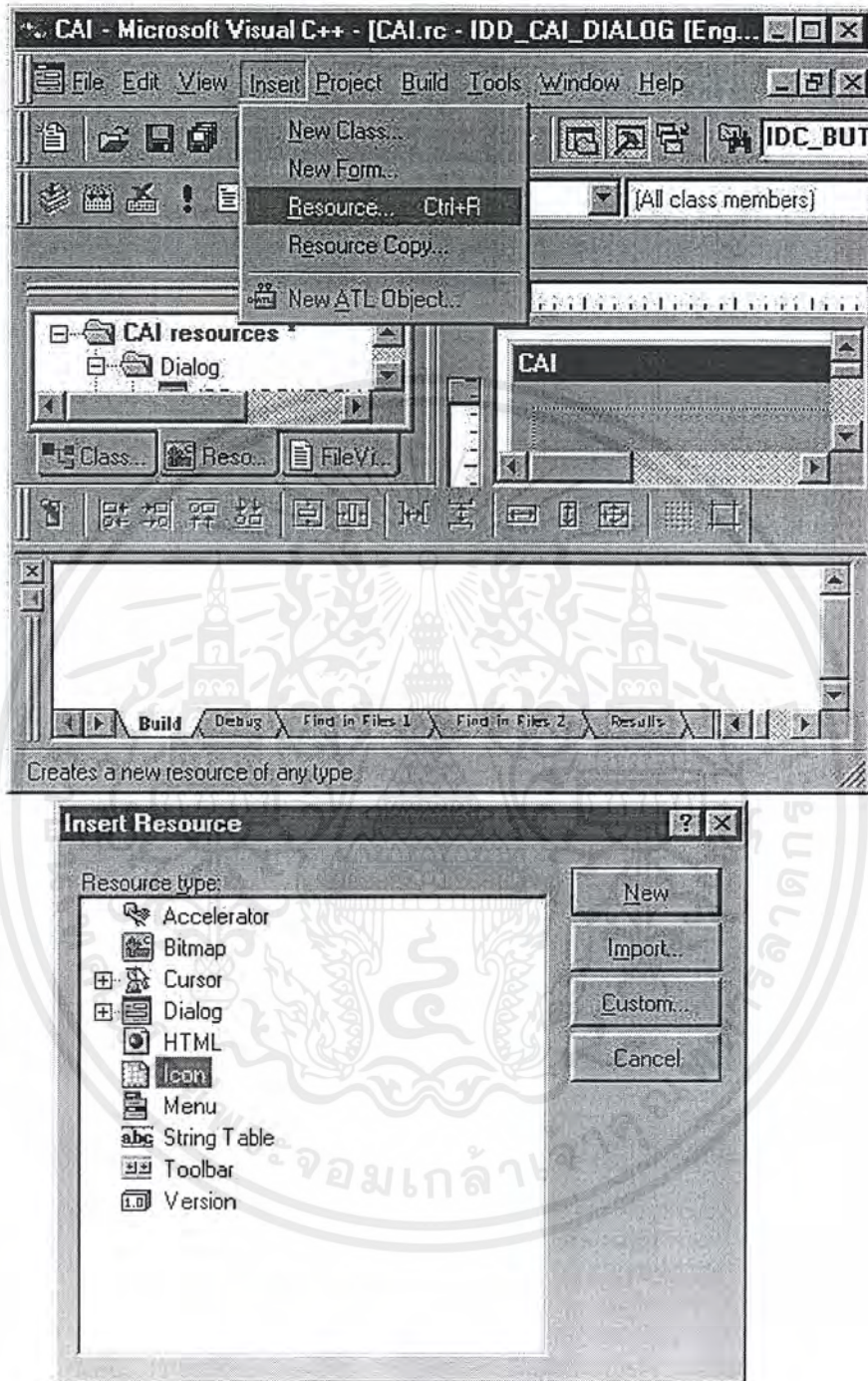
เมื่อสร้างไอคอนริชอร์ส หรือริชอร์สใหม่ โปรแกรมจะกำหนดค่า ID ให้กับริชอร์สที่สร้างโดยอัตโนมัติ ดังที่ได้สร้างไอคอนริชอร์สนี้โปรแกรมได้กำหนดค่า ID ให้เป็น IDI_ICON1 ให้เปลี่ยนค่า ID ของไอคอน นี้ให้เป็นค่าอื่นที่สามารถจำได้ง่ายๆ โดยจะเปลี่ยนจาก IDI_ICON1 ให้เป็น IDI_CPPLOGO

การเปลี่ยนค่า ID ของริชอร์สทำได้โดยคลิกเมาส์ ปุ่มขวาที่ไอคอนริชอร์สนี้จากนั้นเลือกไปที่คำสั่ง Properties ก็จะพบกับไดอะล็อก Icon Properties ให้เปลี่ยนค่าในช่อง ID จาก IDI_ICON1 ให้เป็น IDI_CPPLOGO ดังรูปที่ 2.31

ให้เลือกคอนโทรล Picture จาก คอนโทรลทูลบาร์ ดังรูปที่ 2.32 คอนโทรล Picture ที่นำมาวางในไดอะล็อกนี้ ก็เป็นสแตติกคอนโทรลประเภทหนึ่งเช่นเดียวกัน

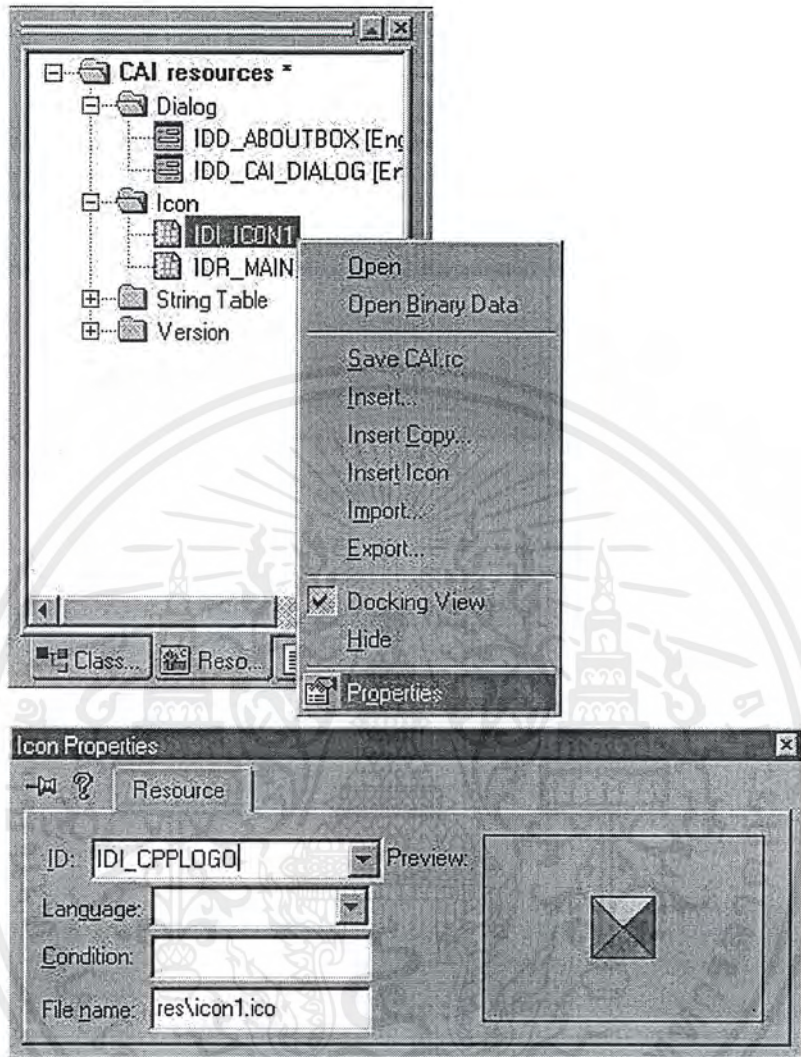
เมื่อวางคอนโทรล Picture เรียบร้อยแล้ว ให้คลิกเมาส์ ปุ่มขวาเพื่อเรียกไดอะล็อก Picture Properties

จากไดอะล็อก Picture Properties ให้กำหนดตัวเลือกในช่อง Type ไปที่ Icon และกำหนดตัวเลือกในช่อง Image ไปที่ IDI_CPPLOGO ดังรูปที่ 2.33 จะเห็นได้ว่าคอนโทรล Picture ที่นำไป วางลงในไดอะล็อกได้เปลี่ยนเป็นรูปไอคอน IDI_CPPLOGO



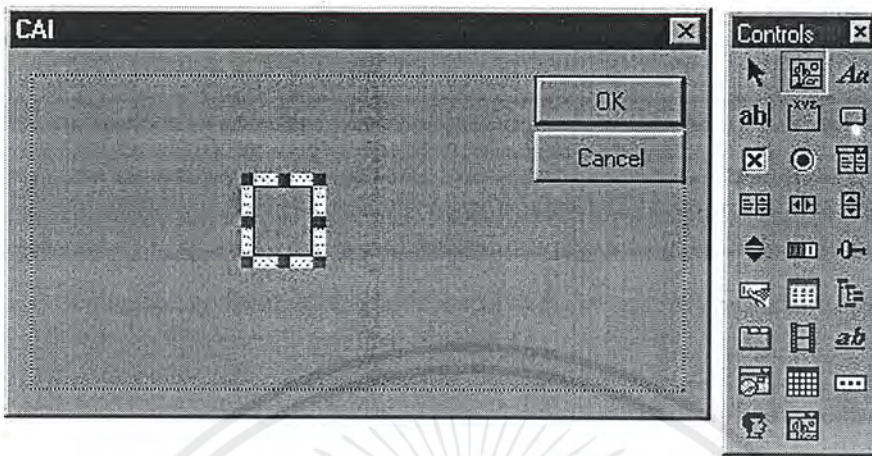
รูปที่ 2.30 การสร้างไอคอนริชอร์สใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

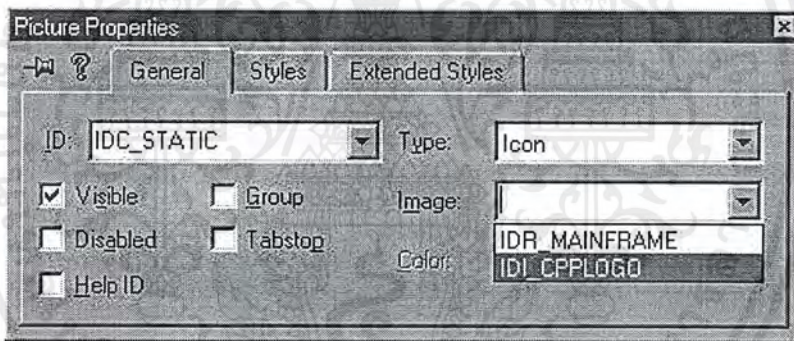


รูปที่ 2.31 การกำหนดค่า ID ให้กับไอคอนรีซอร์ส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



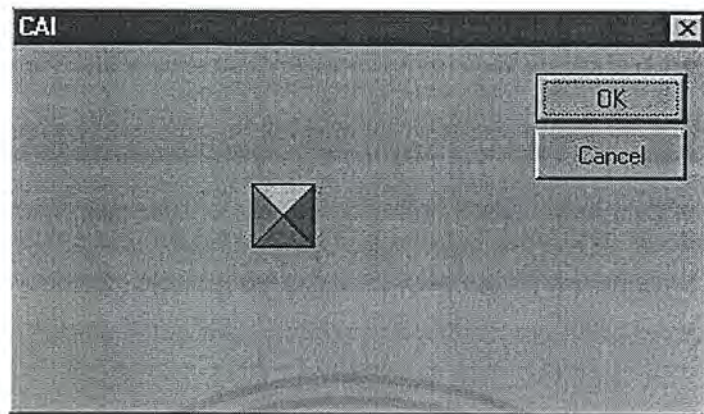
รูปที่ 2.32 การสร้างสแตติกคอนโทรลเพื่อแสดงรูปไอคอน



รูปที่ 2.33 การกำหนดรูปภาพให้กับคอนโทรล

เมื่อลองทดสอบไอคอนโดยกด <Ctrl+T> หรือทดลอง Build โปรเจกต์ และรันโปรแกรม ก็จะได้ผลการรันโปรแกรมดังรูปที่ 2.34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.34 ผลการรันโปรแกรมแสดงรูปไอคอน

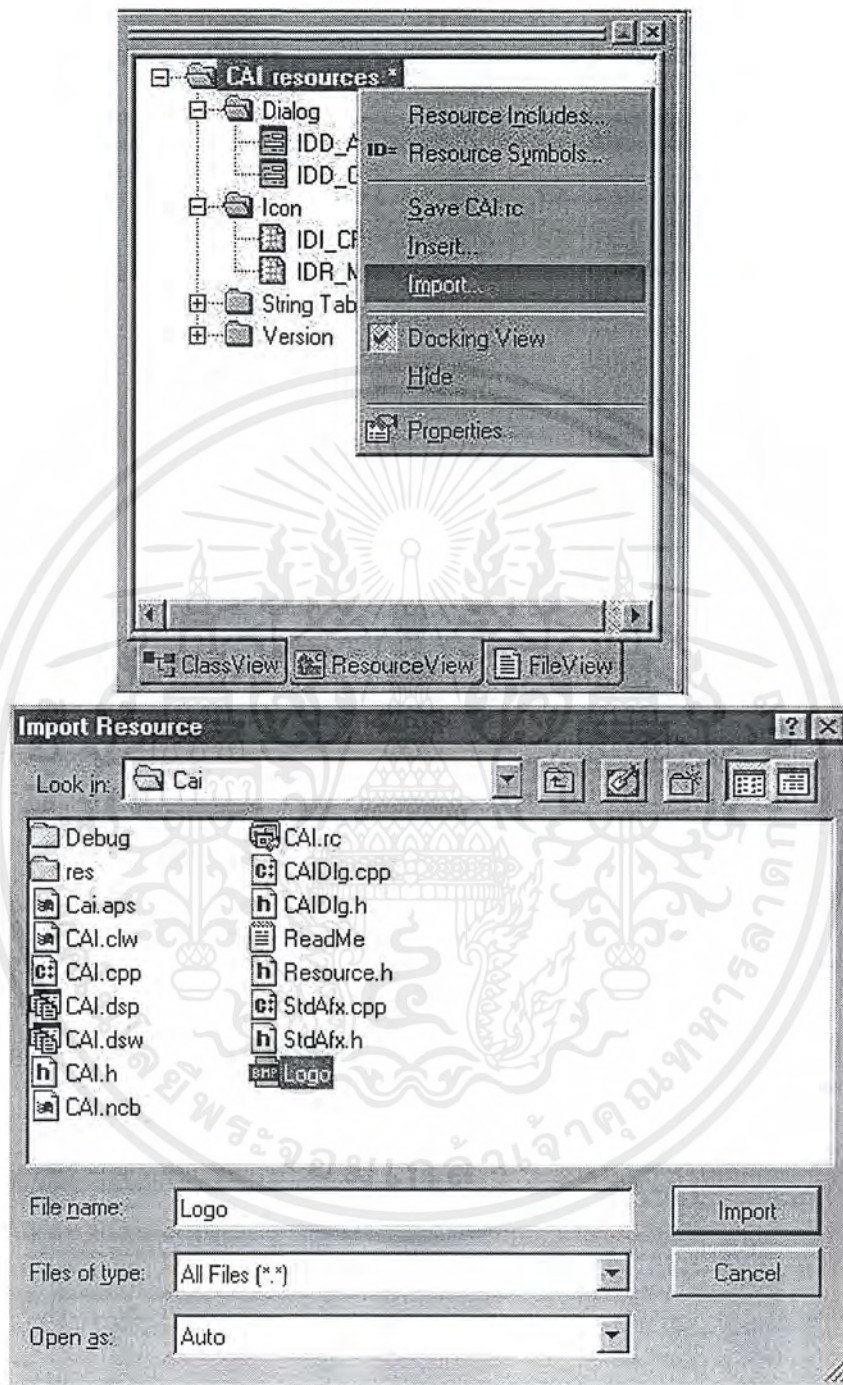
2.10.21 การแสดงรูปภาพที่เป็นไฟล์รูปภาพ

รูปภาพที่สร้างมาแล้วโดยโปรแกรมอื่นสามารถนำมาใส่ลงในไดอะล็อกได้ โดยการ Import รีซอร์สที่มีอยู่แล้วเข้ามาใช้ในโปรเจกต์ซึ่งทำได้โดยการคลิกเมาส์ ปุ่มขวาที่ Resource ในแท็บ Resource เพื่อเรียกเมนูป๊อปอัพ จากนั้นเลือกไปที่คำสั่ง Import ก็จะพบกับไดอะล็อก Import Resource

ให้เลือกรูปภาพ .BMP ที่ต้องการจากไดอะล็อก Import Resource หรืออาจเลือก ไฟล์ .ICO ที่มีอยู่แล้วมาใช้ก็ได้โดยให้เลือกรหัสของไฟล์ไปยัง .ICO และ .BMP หรือที่เหมาะสม และกดปุ่ม Import ดังในรูปที่ 2.35

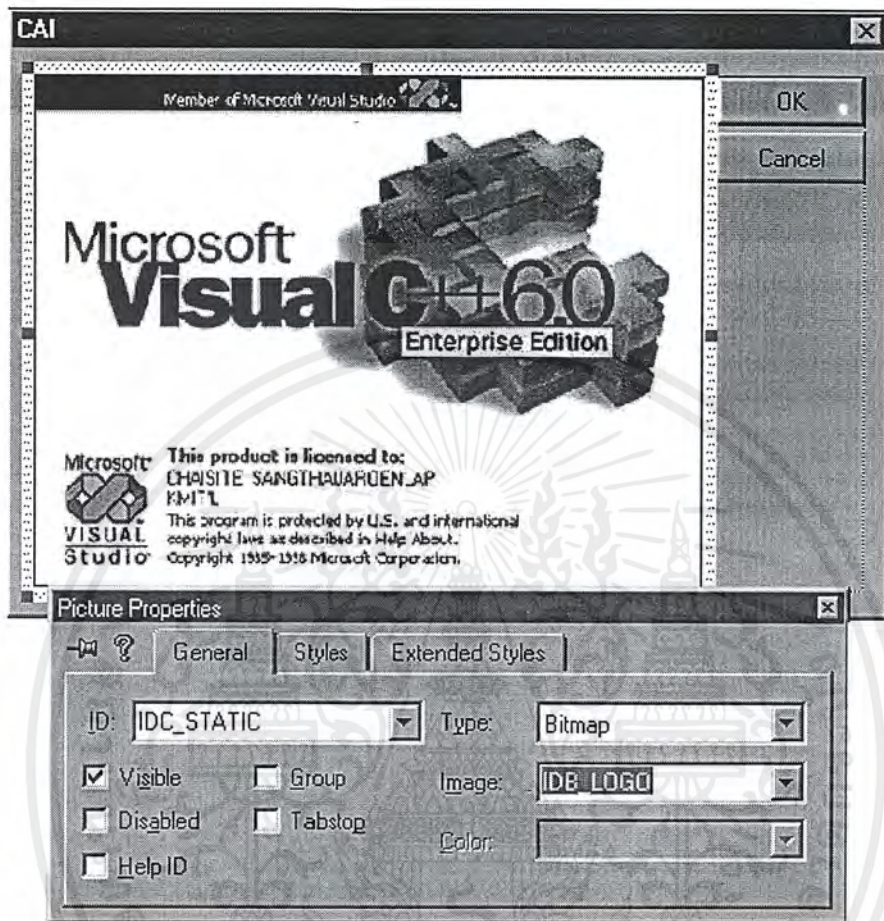
ทำการกำหนด ID ของบิตแมปรีซอร์สที่ Import เข้ามาใหม่นี้ โดยกดปุ่ม <Alt+Enter> เพื่อเรียกไดอะล็อก Bitmap Properties และเปลี่ยนค่าในช่อง ID ให้เป็น IDB_LOGO

จากนั้นให้สร้างคอนโทรล Picture ลงในไดอะล็อกเช่นเดิม และ กำหนด Properties ของคอนโทรลนี้ ให้เป็นแบบ Bitmap และเลือกรูปภาพไปที่ DB_LOGO ดังรูปที่ 2.36 แสดงไดอะล็อกเมื่อได้ Import ไฟล์ IDB_LOGO มาเป็นพื้นหลังของไดอะล็อก



รูปที่ 2.35 การ Import รูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.36 การกำหนดรูปภาพให้กับไดอะล็อก

2.10.22 การ Export รูปภาพ

การ Import ก็มีลักษณะเช่นเดียวกับการโหลดรูปภาพมาใช้เป็นรีซอร์สในโปรเจกต์ แต่การ Export นี้จะเป็นการบันทึกรีซอร์สที่ต้องการเพื่อนำไปใช้ในโปรเจกต์อื่น หรือสำรองข้อมูล คล้ายกับการ Save AS นั่นเอง

สามารถ Export ไอคอนรีซอร์สให้เป็นไฟล์ไอคอน (. ICO) หรือ Export บิตแมปรีซอร์ส ให้เป็นไฟล์รูปภาพ โดยสามารถกำหนดให้เป็นฟอร์แมต .BMP, .DIB, .JPG, JPE และ JPEG ได้ตามต้องการ วิธีการ Export ทำได้โดยคลิกเมาส์ปุ่มขวาที่รีซอร์สที่ต้องการจากนั้น เลือกไปที่คำสั่ง Export

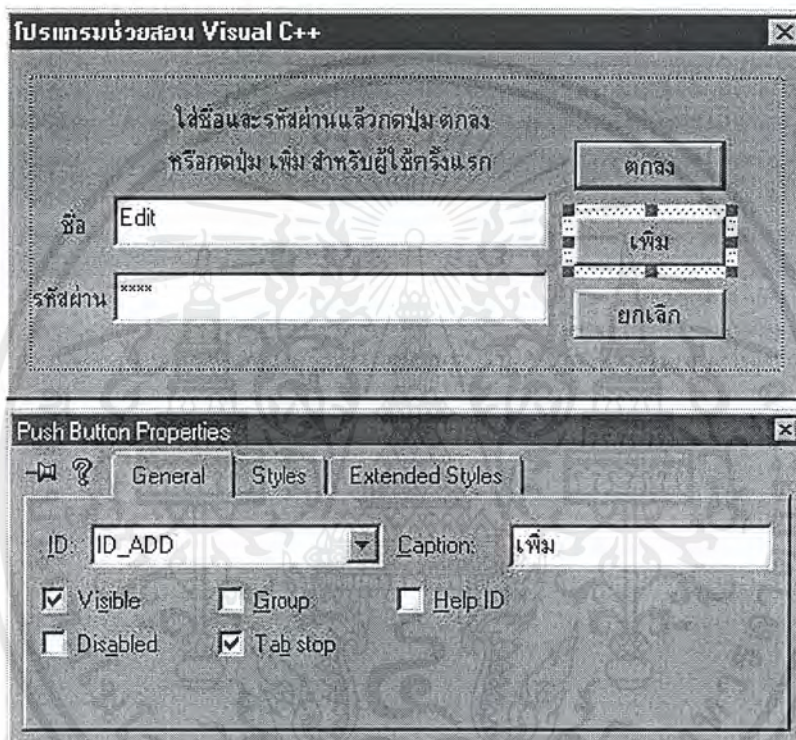
ในการกำหนดนามสกุลหรือฟอร์แมตของไฟล์รูปภาพที่จะ Export นี้ให้พิมพ์นามสกุล ของไฟล์ลงไปได้ทันทีที่ต้องการให้เป็นฟอร์แมตใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.23 การสร้างปุ่มกดในไดอะล็อก

การสร้างปุ่มกดสามารถทำได้โดยการคลิกที่คอนโทรลปุ่มกด แล้วจึงคลิกที่ไดอะล็อกที่ต้องการวางปุ่มกด โดยสามารถปรับแต่งขนาดได้ตามความต้องการ

การแก้ไขข้อความและรูปแบบของปุ่มกดสามารถทำได้โดยการเลือก Button Properties ดังรูปที่ 2.37



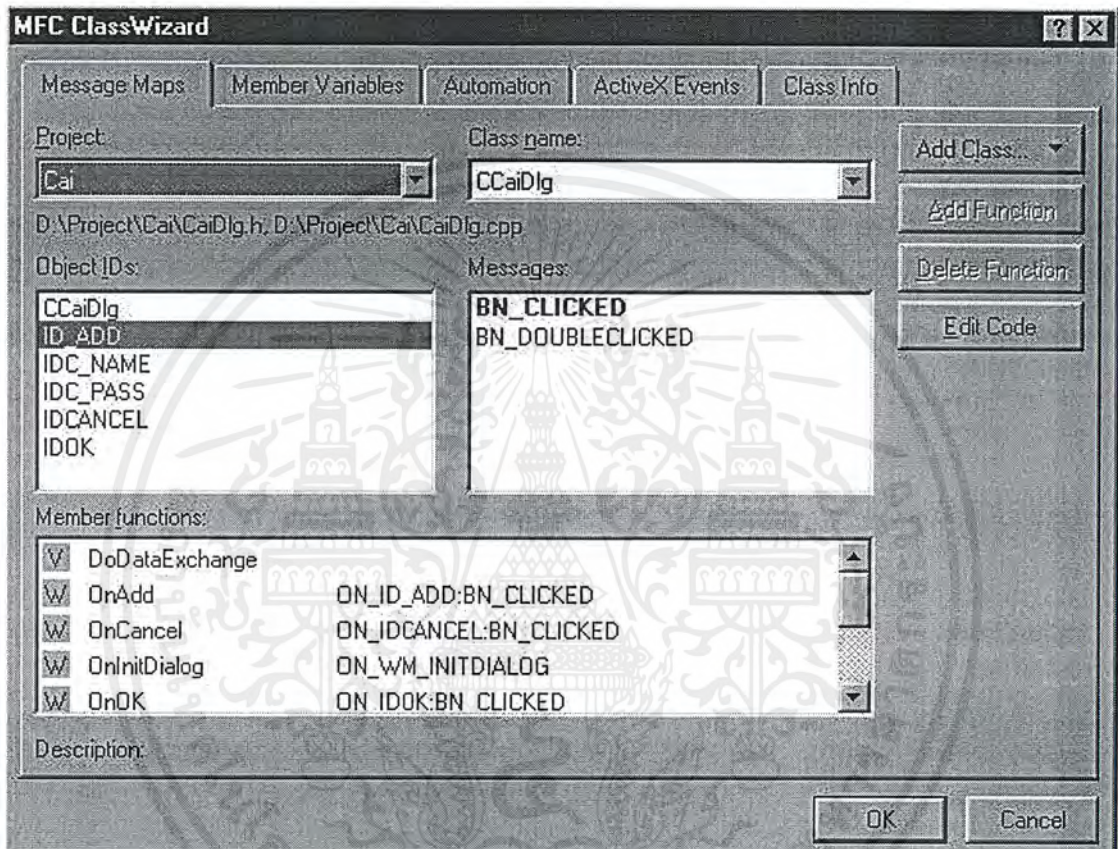
รูปที่ 2.37 การปรับแต่งปุ่มกด

2.10.24 ClassWizard

อุปกรณ์ตัวหนึ่งที่บอกได้ว่าอยู่คู่กับ โปรแกรม Visual C++ 6.0 มาโดยตลอดนอกเหนือจาก AppWizard แล้วก็ยังมี ClassWizard ตัวนี้ที่ทำงานร่วมกันมาตั้งแต่ โปรแกรม Visual C++ 6.0 เวอร์ชัน 1 จนถึง 6 ClassWizard เป็นอุปกรณ์ที่จะช่วยอำนวยความสะดวกในการ จัดการคลาส และการแมปแมสเสจ การสร้างตัวแปรต่างๆ ในกระบวนการ DDX แต่การที่จะสามารถใช้ ClassWizard ได้อย่างมีประสิทธิภาพนั้น ต้องรู้จักกับโครงสร้างของโปรแกรมเหล่านี้พอสมควร

1) ขั้นตอนที่ 1 เรียก ClassWizard

กดปุ่ม <Ctrl+W> หรือเลือกเมนู View ClassWizard จะพบกับไดอะล็อก MFC ClassWizard ดังรูปที่ 2.38



รูปที่ 2.38 ClassWizard

ภายในไดอะล็อก MFC ClassWizard จะเห็นได้ว่ามีแท็บซ้อนๆ กันทั้งหมด 5 แท็บด้วยกัน ทำให้เลือกไปที่แท็บ Message Maps ซึ่งเป็นแท็บแรกทางด้านซ้ายมือสุด และในช่อง Object IDR จะเห็นว่าปุ่ม IDC_ADD รวมอยู่ด้วย ปุ่มนี้ก็คือปุ่ม “เพิ่ม” ที่ได้สร้างไว้ก่อนแล้ว

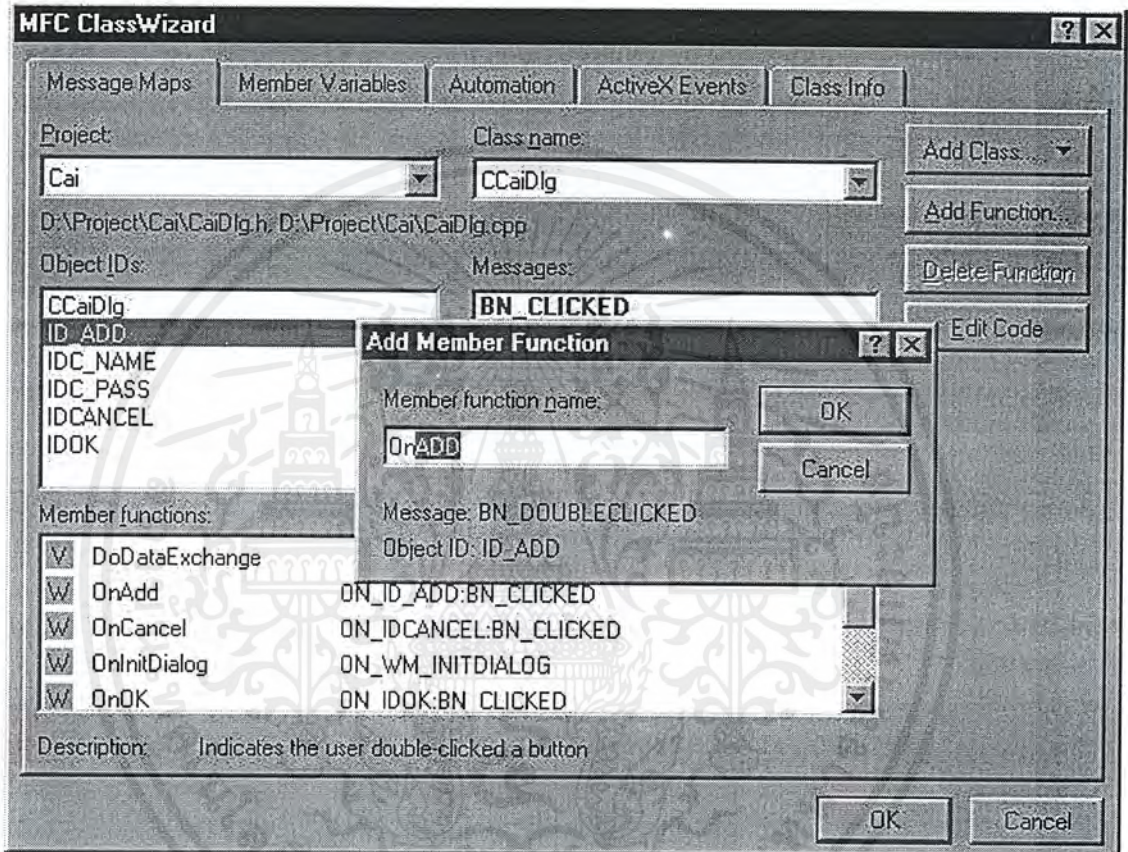
2) ขั้นตอนที่ 2 ทำการแมปปุ่มกด

ให้คลิกไปที่ IDC_ADD จะพบว่า ภายในช่อง Messages ทางด้านขวาจะแสดงรายการแมสเสจของปุ่มกดนี้ โดยมีดังนี้

BN_CLICKED	เป็นแมสเสจของการคลิกปุ่ม
BN_DOUBLECLICKED	เป็นแมสเสจของการดับเบิลคลิก ที่ปุ่ม (ซึ่งไม่นิยม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากต้องการแมสเสจใดของปุ่มกดก็ให้เลือกไปที่แมสเสจนั้น ในที่นี้ ต้องการแมสเสจของการคลิกไปที่คำว่า BN_CLICKED ที่อยู่ในช่อง Messages จากนั้นกดปุ่ม Add Function ดังรูปที่ 2.39



รูปที่ 2.39 การเม็ปแมสเสจปุ่มกด

เมื่อคลิกไปที่ Add Function จะพบว่า มีไอคอน Add member Function เล็กขึ้นมาในไอคอนนี้ เป็นไอคอนที่ใช้สำหรับเพิ่มแมมเบอร์ฟังก์ชันตัวใหม่ให้กับคลาส โดยจะกำหนดชื่อฟังก์ชันให้เป็น OnADD () โดยอัตโนมัติให้กดปุ่ม OK ต่อไป ClassWizard ก็จะสร้างฟังก์ชัน OnADD () ขึ้นมาให้

ฟังก์ชัน OnADD () ที่เพิ่มลงไปนี้จะป็นฟังก์ชันที่ใช้รองรับการกดปุ่ม IDC_ADD หรือปุ่ม “เพิ่ม” โดยเฉพาะ หากต้องการให้มีการกระทำใดๆ เกิดขึ้นเมื่อปุ่มนี้มีการถูกกด เขียนโค้ดโปรแกรมใส่เอาไว้ในฟังก์ชัน OnADD () ของคลาส CbuttonDlg นั้นเอง

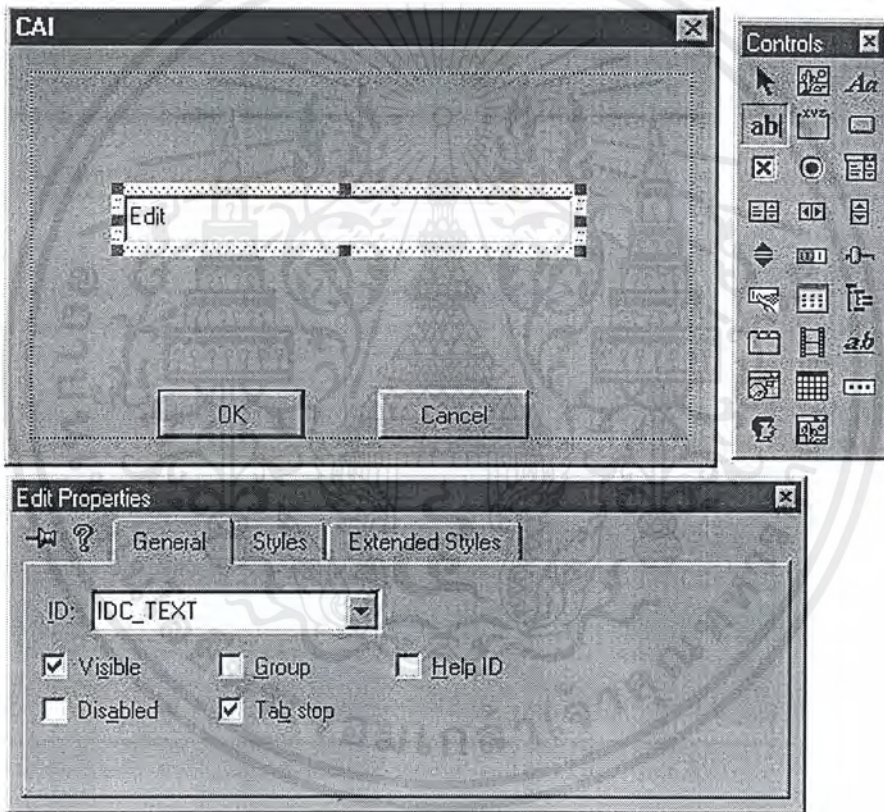
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ขั้นตอนที่ 3 เขียนโค้ดโปรแกรมให้กับฟังก์ชัน

ทำการเขียน โค้ด โปรแกรมตามความต้องการ

2.10.25 การใช้งานแท็กบ็อกซ์บรรทัดเดียว

การสร้าง Text Box สามารถทำได้โดยการนำ Text Box จากทูลบาร์คอนโทรลมาวางในไดอะล็อกโดยปรับให้มีความสูงประมาณ 1 บรรทัด ซึ่งเป็นขนาดปกติ และกดปุ่ม <Alt+Enter> เพื่อเรียกไดอะล็อก Edit Properties ให้เปลี่ยนค่า ID ของคอนโทรลจาก IDC_EDIT1 ให้เป็น IDC_TEXT ดังรูปที่ 2.40



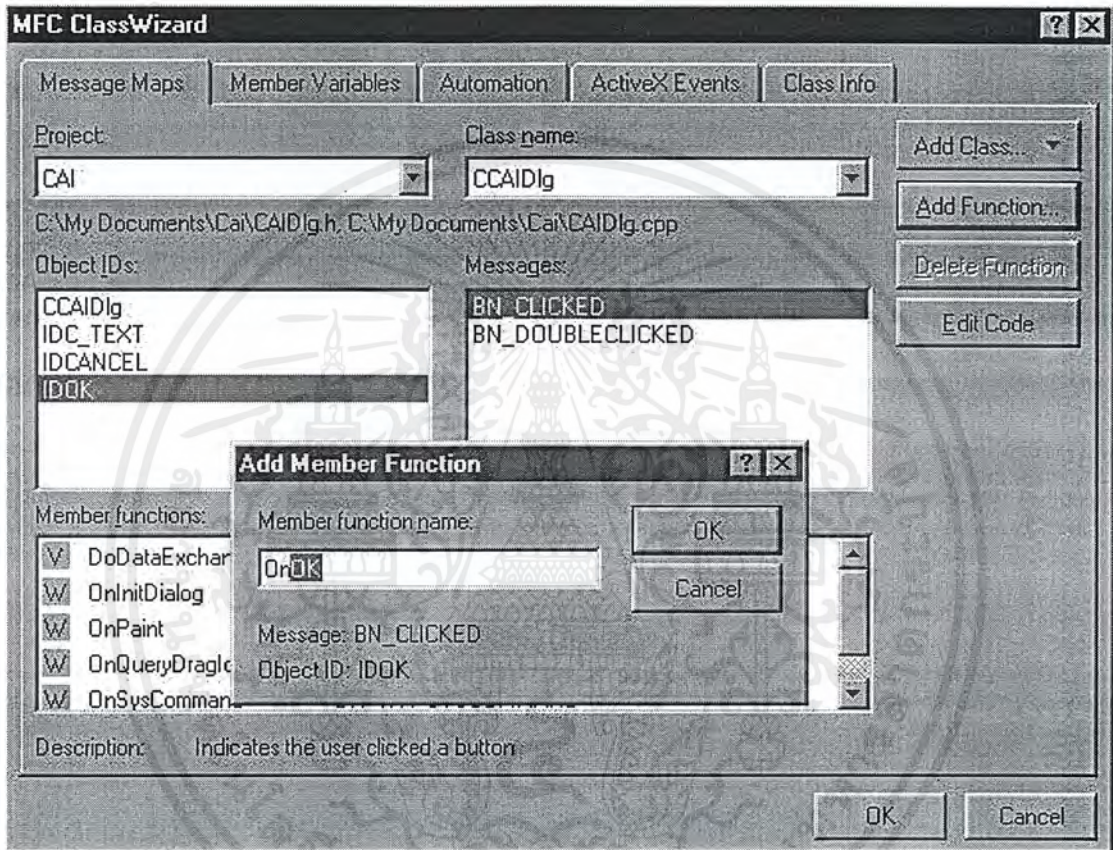
รูปที่ 2.40 การสร้าง Text Box บรรทัดเดียว

2.10.26 การจัดการข้อความในแท็กบ็อกซ์

การจัดการข้อความในแท็กบ็อกซ์ ก็คือการนำข้อความที่อยู่ภายในคอนโทรลแท็กบ็อกซ์ไปใช้ สามารถทำได้หลายวิธี เพราะมีฟังก์ชันที่ใช้ในการจัดการกับ ข้อความในคอนโทรลแท็กบ็อกซ์หลายตัวด้วยกัน จะมาเขียนโปรแกรมจัดการกับข้อความในแท็กบ็อกซ์กัน โดยเมื่อมีการกดปุ่ม Ok เกิดขึ้น ให้มีเมสเสจบ็อกซ์ปรากฏ แสดงข้อความที่อยู่ภายในแท็กบ็อกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งแรกที่จะต้องทำก็คือจะต้องแมปเมสเสจที่ชื่อ BN_CLICKED ให้กับปุ่มกด IDOK โดยให้เรียก ClassWizard ขึ้นมาโดยกด <Ctrl+W> จากนั้นคลิกไปที่ปุ่ม IDOK และทำการแมปกับเมสเสจ BN_CLICKED ดังรูปที่ 2.41



รูปที่ 2.41 การแมปเมสเสจให้กับปุ่มกด

ฟังก์ชัน OnOK() ที่ ClassWizard สร้างขึ้นมาให้ที่นี่ จะเป็นฟังก์ชันที่ใช้รองรับการกดปุ่ม OK ให้อัตโนมัติที่ฟังก์ชัน OnOK() ดังรูปที่ 2.42

```
void CsingleLineDlg::OnOK(){
    //TODO: Add extra validation here
    Cdialog::OnOK();
}
```

รูปที่ 2.42 ฟังก์ชันที่ ClassWizard สร้างขึ้น

จะเห็นว่าภายในฟังก์ชัน บรรทัดสุดท้ายได้เรียก Cdialog::OnOK() ซึ่งเป็นการตอบรับไดอะล็อก เพราะฉะนั้น เมื่อกดปุ่ม OK ก็จะเหมือนกับการปิดไดอะล็อกโดยอัตโนมัติ หากไม่ต้องการให้กดปุ่ม Ok แล้วจบโปรแกรม ก็จะต้องคอมเมนต์บรรทัด Cdialog::OnOK() ดังรูปที่ 2.43

```
void CsingleLineDlg::OnOK(){
    //TODO:Add extra validation here
    // Cdialog::OnOK();
}
```

รูปที่ 2.43 การคอมเมนต์คำสั่ง

เพิ่มโค้ดโปรแกรมลงไปในฟังก์ชัน OnOK() โดยโค้ดโปรแกรมที่จะเพิ่มเข้าไปนี้เป็นการเรียกฟังก์ชันให้อ่านข้อความที่อยู่ภายในเท็กซ์บ็อกซ์ และแสดงออกทางจอภาพโดยใช้ MessageBox ให้แก้ไขฟังก์ชัน CsingleLineDlg::OnOK ให้เป็นดังรูปที่ 2.44

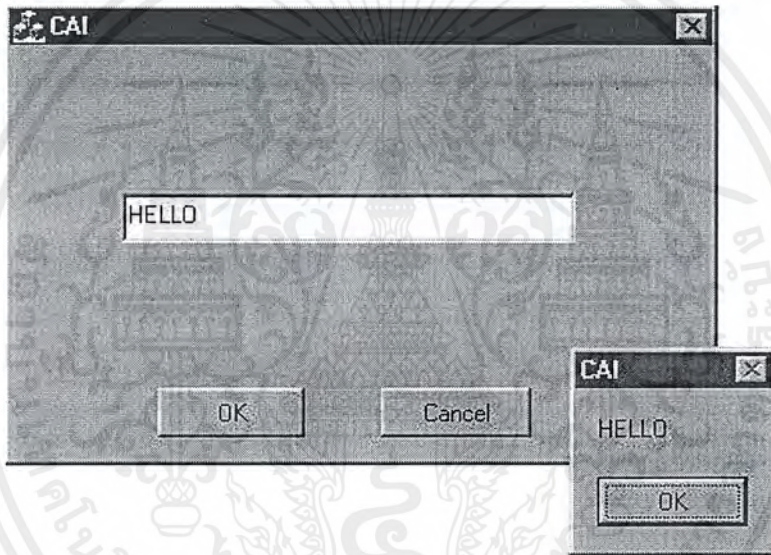
```
void CsingleLineDlg::OnOK(){
    char data [20];
    GetDlgItemText(IDC_TEXT,data,20);
    MessageBox(data);
    Cdialog::OnOK();
}
```

รูปที่ 2.44 การเพิ่มโค้ดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้น ให้ทดลอง Build โปรเจกต์และทดลองรันโปรแกรม เพื่อดูผลการทำงานรูปที่ 2.39 เป็นผลการรันโปรแกรม

เมื่อรันโปรแกรม ให้ลองพิมพ์ข้อความอะไรก็ได้ ลงในช่อง Enter Text และกดปุ่ม OK จะพบกับไดอะล็อกแมสเสจบ็อกซ์ปรากฏขึ้น และแสดงข้อความที่อยู่ในช่องเท็กซ์บ็อกซ์ออกมา ดังในรูป แต่เมื่อกดปุ่มOK ที่แมสเสจบ็อกซ์แล้ว จะเห็นได้ว่า โปรแกรมจะจบการทำงานทันที นั่นเป็นเพราะว่าในฟังก์ชัน OnOK() ที่แก้ไขไปในบรรทัดสุดท้ายได้มีการเรียกฟังก์ชัน Cdialog::OnOK() นั่นเอง



รูปที่ 2.45 ผลการรันโปรแกรม Text Box บรรทัดเดียว

คราวนี้ ให้ลองแก้ไขโปรแกรมอีกครั้งหนึ่ง โดยคอมเมนต์บรรทัด Cdialog::OnOK() เอาไว้

ดังนี้

```

void CsingleLineDlg::OnOK(){
    char data [50];
    GetDlgItemText(IDC_TEXT,data,50);
    MessageBox(data);
    //Cdialog::OnOK();
}

```

รูปที่ 2.46 การใช้คำสั่ง MessageBox()

จากนั้นให้ Build โปรเจกต์ใหม่อีกครั้ง และทดลองรันโปรแกรม จะพบว่าเมื่อกดปุ่ม OK ที่ไดอะล็อกแมสเสจบ็อกซ์ โปรแกรมจะไม่จบการทำงานอีกแล้ว ยังสามารถใช้โปรแกรมต่อไปได้ ถ้าหากต้องการจบโปรแกรม ให้กดปุ่ม Cancel หรือกด <ESC> แทนโปรแกรมก็จะจบการทำงาน

2.10.27 ทำความเข้าใจกับโค้ดโปรแกรม

จะมาทำความเข้าใจกับโปรแกรมที่ได้เขียนไปที่ละบรรทัด โดยเริ่มจากบรรทัดแรกซึ่งเป็นการกำหนดตัวแปรอะเรย์แบบ char ที่ชื่อ data มีขนาด 50 ตัวอักษร จะใช้ตัวแปรนี้เก็บข้อความที่พิมพ์ลงไปในพื้นที่บ็อกซ์

```
Char data [50];
```

รูปที่ 2.47 การประกาศตัวแปร

บรรทัดที่ 2 เป็นการเรียกฟังก์ชัน GetDlgItemText() เพื่อนำข้อมูลที่อยู่ในเทกบ็อกซ์มาเก็บไว้ในตัวแปร data พารามิเตอร์ก็คือค่า ID ของเทกซ์บ็อกซ์ พารามิเตอร์ตัวที่ 2 ก็คือตัวแปรที่จะใช้เก็บค่า ส่วนพารามิเตอร์ตัวสุดท้ายก็คือ จำนวนความยาวของตัวอักษรสูงสุดที่จะอ่านเข้ามา รายละเอียดของฟังก์ชันนี้ สามารถค้นหาได้จากระบบช่วยเหลือของ โปรแกรม Visual C++ 6.0

```
GetDlgItemText(IDC_TEXT,data,50);
```

รูปที่ 2.48 ฟังก์ชัน GetDlgItem()

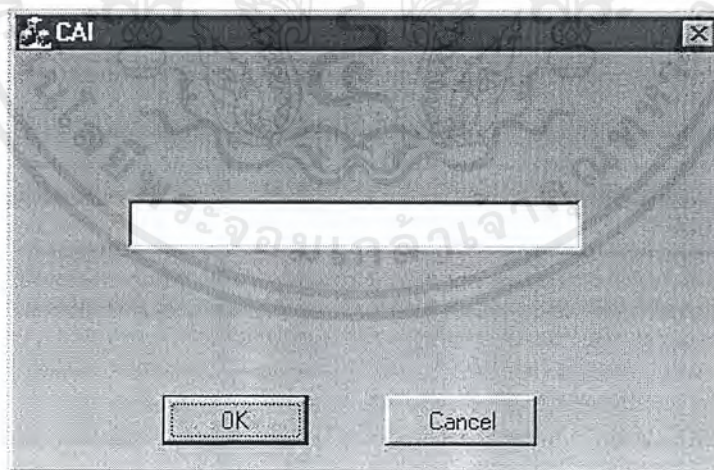
จากนั้น ในบรรทัดสุดท้ายเป็นการเรียกฟังก์ชัน MessageBox () เพื่อแสดงข้อความที่อยู่ในตัวแปร data

```
MessageBox(data);
```

รูปที่ 2.49 ฟังก์ชัน MessageBox()

2.10.28 การเปลี่ยนลำดับโฟกัส

จากการทำงานของโปรแกรมในข้างต้น ให้สังเกตโปรแกรมในตอนทำงาน จะพบว่าปุ่ม OK ถูกโฟกัสเป็นอันดับแรก ดังรูปที่ 2.50



รูปที่ 2.50 การทำงานของโปรแกรม Text Box บรรทัดเดียว

ลำดับการโฟกัสของคอนโทรลต่างๆในไดอะล็อก ไม่ใช่เรื่องที่จะมองข้ามไปได้ โดยให้นึกถึงสภาพการใช้งานโปรแกรมต่างๆไป เมื่อรันโปรแกรม SingleLine ขึ้นมา เพื่อพิมพ์ข้อมูลลงไป ช่อง Enter Text แต่เคอร์เซอร์กลับไปโฟกัสอยู่ที่ปุ่ม OK ซึ่งผิดวิสัย จึงต้องนำเมาส์ไปคลิกที่ช่องรับเอกสารเป็นเอกสารที่ส่งวันวิชาหรือการแข่งงานเพื่อการศึกษาเท่านั้น เมื่อนักผู้ดูแลเห็นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อความเพื่อให้ปรากฏเคอร์เซอร์พร้อมรับการพิมพ์ข้อความ ตรงจุดนี้ ถ้าจัดลำดับของ Tab Order ให้ดีและเหมาะสมกับการใช้งาน โปรแกรมก็จะใช้งานได้ง่าย และทำให้น่าใช้ขึ้นมา

การเปลี่ยนลำดับการโฟกัสของคอนโทรลในไดอะล็อก จะมีอยู่ด้วยกัน 2 วิธี วิธีแรกที่จะแนะนำก็คือ การใช้ไดอะล็อกอิดิตเตอร์ปรับลำดับการโฟกัสคอนโทรล โดยให้เรียกไดอะล็อกกริชอร์ส IDD_SINGLELINE_DIALOG ขึ้นมาแก้ไขใหม่ แล้วกดปุ่ม <Ctrl+D> หรือเลือกคำสั่ง Layout> Tab Order จะพบว่ามีตัวเลขปรากฏขึ้นตามคอนโทรลที่อยู่ในไดอะล็อก ให้ใช้การคลิกเมาส์ ไปที่แท็บบ็อกซ์ ซึ่งเป็นคอนโทรลตัวที่ต้องการ โฟกัสเป็นครั้งแรก และคลิกไปยังปุ่มกด OK และ Cancel ตามลำดับ

จากนั้นให้ทำการ Build โปรแกรมใหม่อีกครั้งและทดลองรันโปรแกรมดู จะพบว่าตอนนี้เคอร์เซอร์อยู่ที่แท็บบ็อกซ์แล้ว

ให้ลองกดปุ่ม <Tab> จะเห็นได้ว่า เคอร์เซอร์จะย้ายจากแท็บบ็อกซ์ไปโฟกัสที่ปุ่ม OK และปุ่ม Cancel ตามลำดับ

2.10.29 การตรวจสอบข้อความในแท็บบ็อกซ์

สามารถตรวจสอบข้อความภายในแท็บบ็อกซ์ เพื่อดูว่า得有ข้อความพิมพ์ลงไปเรียบร้อยแล้วหรือไม่ ก่อนที่จะแสดงข้อความนั้นออกทางแมสเสจบ็อกซ์ การตรวจจับข้อความนี้ ให้จำวิธีการใช้งานให้ดี เพราะจะใช้กันมากเมื่อต้องการให้โปรแกรมทำงานอย่างมีประสิทธิภาพ และหลีกเลี่ยงจากการเกิดข้อผิดพลาดในการทำงาน

การตรวจสอบข้อความ จะต้องนำเงื่อนไขเข้ามาใช้ โดยเมื่ออ่านข้อความภายในแท็บบ็อกซ์ มาเก็บไว้ในตัวแปร data แล้ว ในบรรทัดต่อจากนี้ จะต้องใส่โค้ดโปรแกรมเช็คข้อความลงไป ดังรูปที่ 2.51

เมื่อรันโปรแกรม ถ้าไม่พิมพ์ข้อความใดๆ ลงในแท็บบ็อกซ์ (“สตริงใน data มีค่าเป็น”) เมื่อกดปุ่ม OK ก็จะมีแมสเสจบ็อกซ์แสดงข้อความออกมาว่า “Error,no text” แต่ถ้าพิมพ์ข้อความลงไป ก็จะปรากฏแมสเสจบ็อกซ์แสดงข้อความนั้นออกทางจอภาพ เช่นเดิม

```

void CsingleLineDlg::OnOK(){
    char data [50];
    GetDlgItemText(IDC_TEXT,data,50);
    if (!strcmp(data.""))
        MessageBox("Error,no text");
    Else
        MessageBox(data);
}

```

รูปที่ 2.51 โปรแกรมตรวจสอบข้อความในเท็กซ์บ็อกซ์

ฟังก์ชัน strcmp เป็นฟังก์ชันที่ใช้สำหรับเปรียบเทียบสตริง 2 ตัว ถ้าฟังก์ชันคืนค่า 0 กลับมาก็แสดงว่าสตริง 2 ตัวนั้นเหมือนกัน โค้ดโปรแกรมในข้างต้นนั้น สามารถเขียนในอีกลักษณะหนึ่งได้ดังรูปที่ 2.52

```

if (strcmp(data,"")==0)
    MessageBox("Error,no text");
Else
    MessageBox(data);

```

รูปที่ 2.52 โปรแกรมการตรวจสอบข้อความในเท็กซ์บ็อกซ์อีกแบบหนึ่ง

2.10.30 คลาสที่ใช้ในการจัดการกับเท็กซ์บ็อกซ์

ในการสร้างคอนโทรลหลายๆประเภท จะมีคลาสที่ใช้ในการจัดการกับคอนโทรลโดยเฉพาะ ยกตัวอย่างเช่น การสร้างข้อความโดยใช้สแตติกคอนโทรล ก็จะมีคลาส CStatic ที่ใช้สำหรับควบคุมการทำงานของสแตติกคอนโทรล สำหรับเท็กซ์บ็อกซ์ก็เช่นเดียวกัน คลาสที่ใช้ในการควบคุมเท็กซ์บ็อกซ์ก็คือคลาส CEdit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาส CEdit เป็นคลาสหนึ่งที่สืบทอดมาจากคลาส CWnd ของ MFC ซึ่งคลาส CWnd นี้เป็นคลาสแม่ของคอนโทรลคลาสทั้งหมด ฟังก์ชัน GetDlgItem() ก็เป็นฟังก์ชันที่สืบทอดมาจากคลาส CWnd นี้เอง

ภายในคลาส CEdit ได้เตรียมฟังก์ชันเอาไว้มากมาย ให้สามารถควบคุมการทำงานของแท็กซ์บ็อกซ์ได้ ฟังก์ชันที่น่าสนใจมีดังต่อไปนี้

- 1) SetSel ใช้สำหรับสร้างแถบสีล้อมรอบข้อความในแท็กซ์บ็อกซ์ (เลือกข้อความ)
- 2) Clear ใช้สำหรับลบข้อความที่ถูกเลือกภายในแท็กซ์บ็อกซ์
- 3) Cut ลบข้อความที่ถูกเลือกภายในแท็กซ์บ็อกซ์ และก๊อปปี้ลงคลิปบอร์ด
- 4) Copy ก๊อปปี้ข้อความที่ถูกเลือกภายในแท็กซ์บ็อกซ์ลงคลิปบอร์ด
- 5) Paste คัดลอกข้อความภายในคลิปบอร์ดลงแท็กซ์บ็อกซ์
- 6) Undo ยกเลิกการแก้ไขครั้งสุดท้ายของข้อความภายในแท็กซ์บ็อกซ์
- 7) EmptyUndoBuffer เคลียร์หน่วยความจำที่ใช้ในการ Undo
- 8) LimitText จำกัดจำนวนตัวอักษรที่สามารถพิมพ์ลงไปในแท็กซ์บ็อกซ์ได้
- 9) ReplaceSel แทนที่ข้อความที่ถูกเลือกภายในแท็กซ์บ็อกซ์โดยใช้ข้อความที่ต้องการ
- 10) SetReadOnly กำหนดให้แท็กซ์บ็อกซ์สามารถอ่านได้อย่างเดียว
- 11) SetPasswordChar กำหนดค่าตัวอักษรที่จะแสดงในแท็กซ์บ็อกซ์แบบ password
- 12) GetPasswordChar คืนค่าตัวอักษรที่แสดงในแท็กซ์บ็อกซ์แบบ password

2.10.31 การสร้างออบเจกต์คอนโทรล

เมื่อสร้างคอนโทรลโดยใช้ไดอะล็อกอิดิตเตอร์ ซึ่งเป็นการสร้างคอนโทรลที่ต่างจากการเขียน โค้ดโปรแกรมด้วยตนเอง การเรียกใช้ฟังก์ชันจากคลาส จึงเรียกใช้ใดๆ เลยไม่ได้ แต่ต้องใช้ออบเจกต์ของคลาสนั้นในการเรียก

จากแท็กซ์บ็อกซ์ที่ได้ทำการสร้างไปนั้น สามารถสร้างออบเจกต์ของคอนโทรลได้โดยเขียนโค้ดโปรแกรมดังรูปที่ 2.53

```
CEdit *textbox;
textbox = (CEdit*)GetDlgItem(IDC_TEXT);
```

รูปที่ 2.53 การสร้างออบเจกต์คอนโทรล

จากโค้ดโปรแกรมในข้างต้น จะได้ตัวแปร `textbox` เป็นออบเจกต์ของคอนโทรล `IDC_TEXT` ซึ่งจะนำไปใช้ในการเรียกฟังก์ชันต่อไป เช่นเรียกฟังก์ชัน `SetFocus` เพื่อใช้กำหนดเคอร์เซอร์ให้มาโฟกัสที่เท็กซ์บ็อกซ์ จะต้องเขียนโค้ดโปรแกรมดังรูปที่ 2.54

```
textbox->SetFocus();
```

รูปที่ 2.54 การใช้ฟังก์ชัน `SetFocus()`

ที่ต้องใช้เครื่องหมาย `->` ก็เพราะว่า ได้ประกาศตัวแปร `textbox` ซึ่งเป็นออบเจกต์ของคลาส ให้เป็นแบบ `Pointer` ที่ต้องประกาศให้เป็นแบบ `Pointer` นั่นก็เพราะว่า จะต้องใช้ความสามารถของตัวแปร `Pointer` ชี้ไปที่คอนโทรลตัวที่ต้องการในไดอะล็อก เช่น `IDC_TEXT` เป็นต้น

2.10.32 การเลือกข้อความภายในเท็กซ์บ็อกซ์

การเลือกข้อความภายในเท็กซ์บ็อกซ์ หมายถึง การทำแถบสีล้อมรอบข้อความที่อยู่ภายในช่อง จะใช้ฟังก์ชัน `SetSel()` ในการเลือกข้อความ (`SetSel` ย่อมาจากคำว่า `Set Select`) ซึ่งมีรูปแบบดังรูปที่ 2.55

```
void SetSel (int nStartChar,intnEndChar,BOOI bNoScroll = FALSE);
```

รูปที่ 2.55 รูปแบบของคำสั่ง `SetSel()`

พารามิเตอร์ที่จะต้องใส่ให้กับฟังก์ชันนี้ก็คือ

- 1) `nStartChar` ตำแหน่งเริ่มต้นของตัวเลือก
- 2) `nEndChar` ตำแหน่งสิ้นสุดของตัวเลือก
- 3) `bNoScroll` ให้มีการสกรอลล์ไปยังตัวอักษรที่อยู่เลยกรอบหรือไม่ โดยปกติจะตั้งค่าไว้ที่ `FALSE`

ตัวอย่างเช่น หากต้องการเลือกข้อความภายในเท็กซ์บ็อกซ์ ตั้งแต่อักษรตัวแรกถึงตัวที่ 5 จะต้องเขียนโปรแกรมดังรูปที่ 2.56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void CsingleLineDlg::OnOK(){
    CEdit*textbox;
    textbox =(CEdit*)GetDlgItem(IDC_TEXT);
    textbox ->SetSel(0,5);
    textbox->SetFocus();
}

```

รูปที่ 2.56 ตัวอย่างการใช้ฟังก์ชัน SetSel()

ให้แก้ไขโค้ดโปรแกรมในฟังก์ชัน OnOk() ดังที่แสดงในตัวอย่างข้างต้น และทำการ Build เพื่อรันโปรแกรม เมื่อรันโปรแกรมให้ใส่ตัวอักษรลงไปสัก 10 ตัว จากนั้นกดปุ่ม OK จะพบว่าเกิดแถบสีล้อมรอบข้อความภายในเท็กซ์บ็อกซ์ โดยเริ่มตั้งแต่ตัวอักษรตัวแรกไปจนถึงตัวที่ 5

จากผลการรันโปรแกรมในรูปที่ 2.57 และโค้ดโปรแกรมในข้างต้น จะเห็นได้ว่า ได้เรียกฟังก์ชัน SetSel โดยกำหนดค่าเริ่มต้นให้เป็น 0 หมายถึงตัวอักษรตัวแรกสุดและกำหนดค่าสิ้นสุดไว้ที่ 5 เพราะฉะนั้นจึงเกิดแถบสีตั้งแต่อักษรตัวที่ 1 ไปจนถึงตัวที่ 5



รูปที่ 2.57 ผลการรันการเลือกข้อความ

และจะสังเกตได้อย่างหนึ่งว่า เมื่อเรียกฟังก์ชัน SetSel แล้ว ได้ทำการเรียกฟังก์ชัน SetFocus เพื่อเป็นการย้ายตำแหน่งเคอร์เซอร์ให้กลับมาโฟกัสที่เท็กซ์บ็อกซ์เหมือนเดิม เพราะเมื่อกดปุ่ม OK เคอร์เซอร์ก็จะไปโฟกัสที่ปุ่ม OK เพราะฉะนั้น จึงต้องเรียกให้กลับมาที่โฟกัสที่เท็กซ์บ็อกซ์อีกครั้ง

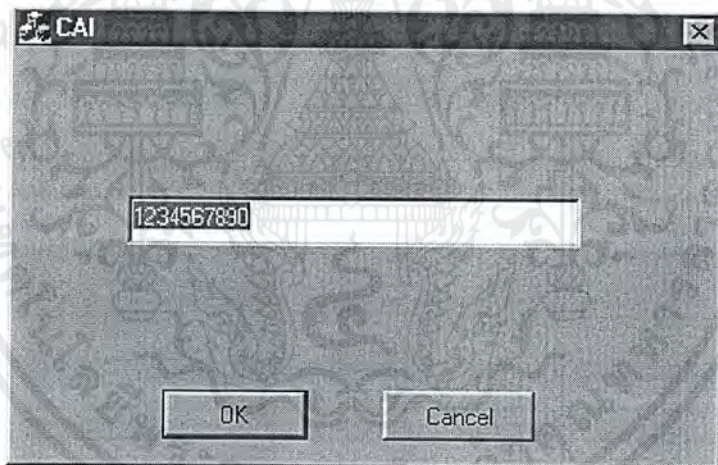
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าต้องการเลือกข้อความทั้งหมดที่อยู่ในเท็กซ์บ็อกซ์ จะกำหนดค่าให้ฟังก์ชัน SetSel เป็น 0 กับ -1 ดังรูปที่ 2.58

```
textbox->SetSel(0,-1);
textbox->SetFocus();
```

รูปที่ 2.58 การเลือกข้อความทั้งหมดในเท็กซ์บ็อกซ์

เมื่อรันโปรแกรม ให้พิมพ์ข้อความใดๆลงไปและกดปุ่ม OK จะพบว่าเกิดแถบสีล้อมรอบข้อความทั้งหมดที่พิมพ์ลงไป เพราะการกำหนดค่า 0 และ -1 ให้กับฟังก์ชัน SetSel() นั้นมีความหมายว่า ให้แสดงแถบสีเลือกตัวอักษรทั้งหมดภายในเท็กซ์บ็อกซ์



รูปที่ 2.59 ผลการรันการเลือกข้อความทั้งหมด

2.10.33 การลบข้อความภายในเท็กซ์บ็อกซ์

การลบข้อความภายในเท็กซ์บ็อกซ์มีอยู่ด้วยกัน 2 วิธีคือ

- 1) การลบแบบธรรมดาโดยใช้ฟังก์ชัน Clear

การลบโดยใช้ฟังก์ชัน Clear จะเป็นการลบข้อความแบบปกติ โดยจะต้องสร้างแถบสีล้อมรอบข้อความที่ต้องการลบเสียก่อน ดังโค้ดโปรแกรกดังรูปที่ 2.60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

textbox->SetSel(0,-1);
textbox->SetFocus();
textbox->Clear();

```

รูปที่ 2.60 การลบข้อความในเท็กซ์บ็อกซ์โดยใช้ฟังก์ชัน Clear()

2) การลบและก๊อปปี้ลงคลิปบอร์ดโดยฟังก์ชัน Cut

การลบข้อความแบบนี้ จะเหมือนกับการลบโดยใช้ฟังก์ชัน Clear แต่ข้อความที่ถูกลบจะก๊อปปี้ลงในคลิปบอร์ดด้วย ดังรูปที่ 2.61

```

textbox->SetSel(0,-1);
textbox->SetFocus();
textbox->Cut();

```

รูปที่ 2.61 การลบข้อความในเท็กซ์บ็อกซ์โดยใช้ฟังก์ชัน Cut()

2.10.34 การใช้งานฟังก์ชัน Copy,PasteและUndo

สามารถก๊อปปี้และคัดลอกข้อความในคลิปบอร์ดได้โดยใช้ฟังก์ชัน Copy และ Pasteซึ่งการทำงานของฟังก์ชันทั้งสองตัวนี้ เหมือนกับการใช้คำสั่ง Copy และ Paste จากเมนู Edit ของโปรแกรมอิดิตเตอร์ต่างๆ ไป ตัวอย่างต่อไปนี้เป็นการใช้งานฟังก์ชัน Copy()

```

textbox->SetSel(0,-1);
textbox->SetFocus();
textbox->Copy();

```

รูปที่ 2.62 การใช้งานฟังก์ชัน Copy()

จากโค้ดโปรแกรมเป็นการก๊อปปี้ข้อความทั้งหมดลงคลิปบอร์ด โดยสามารถทดสอบดูว่าข้อความถูกก๊อปปี้ลงคลิปบอร์ดจริงหรือไม่ โดยการเปิดโปรแกรม NotePad และทำการเลือกคำสั่ง Paste จากเมนู Edit

สำหรับการใช้งานฟังก์ชัน Paste() ทำได้โดยทดลอง Copy ข้อความที่ต้องการลงคลิปบอร์ดก่อนและเรียกฟังก์ชัน Paste ดัง โค้ดโปรแกรมดังรูปที่ 2.63

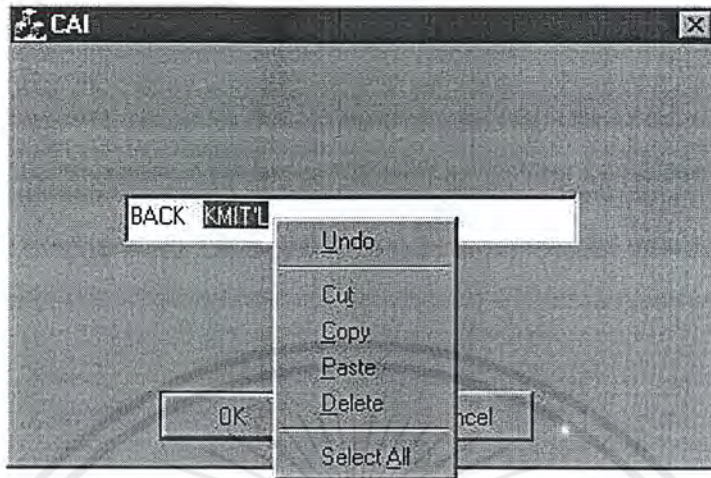
```
CEdit *textbox;
textbox=(CEdit*)GetDlgItem(IDC_TEXT);
textbox->Paste();
```

รูปที่ 2.63 การใช้ฟังก์ชัน Paste()

การ Undo คือการยกเลิกการแก้ไขข้อความครั้งสุดท้ายในเท็กซ์บ็อกซ์ สามารถเรียกใช้ฟังก์ชันนี้ได้เช่นเดียวกับฟังก์ชัน Paste แต่ในการใช้งานจริงแล้ว จะไม่พบการเรียกฟังก์ชันลักษณะนี้บ่อยนัก เพราะสามารถปุ่มฮอตคีย์กับเท็กซ์บ็อกซ์ได้โดยตรง โดยใช้ปุ่มดังนี้

- 1) Ctrl+C เป็นการก๊อปปี้ข้อความที่เลือกไว้ลงคลิปบอร์ด เช่นเดียวกับฟังก์ชัน Copy
- 2) Ctrl+V เป็นการคัดลอกข้อความจากคลิปบอร์ดลงในเท็กซ์บ็อกซ์ เช่นเดียวกับฟังก์ชัน Paste
- 3) Ctrl+Z เป็นการยกเลิกการแก้ไขครั้งสุดท้าย เช่นเดียวกับการเรียกฟังก์ชัน Undo

นอกจากจะสามารถใช้ ปุ่มฮอตคีย์แทนการเรียกฟังก์ชันได้แล้วยังสามารถ เรียกเมนูป๊อปอัพเพื่อจัดการกับข้อความได้อีกด้วย โดยคลิกเมาส์ปุ่มขวาที่เท็กซ์บ็อกซ์ก็จะพบกับป๊อปอัพเมนูที่แสดงคำสั่งต่างๆ



รูปที่ 2.64 เมนูป๊อปอัพ

2.10.35 การแทนที่ข้อความด้วยฟังก์ชัน ReplaceSel

สามารถแทนที่ข้อความภายในเท็กซ์บ็อกซ์ได้ โดยใช้ฟังก์ชัน ReplaceSel (ย่อมาจากคำว่า Replace Select) รูปแบบการใช้งานมีดังรูปที่ 2.65

```
void ReplaceSel(LPCTSTR IpszNewText,BOOL bCanUndo =FALSE);
```

รูปที่ 2.65 รูปแบบการใช้งานฟังก์ชัน ReplaceSel()

พารามิเตอร์ของฟังก์ชัน ReplaceSel มีดังนี้

- 1) IpszNewText คือข้อความใหม่ที่ต้องการแทนที่
- 2) BcanUndo เป็นการกำหนดว่าข้อความที่แทนที่ลงไปจะสามารถ Undo ได้หรือไม่ (ปกติจะเป็น FALSE)

การใช้งานฟังก์ชัน ReplaceSel มีดังรูปที่ 2.66

```

void CsingleLineDlg::OnOK(){
    CEdit *textbox;
    Textbox=(CEdit*)GetDlgItem(IDC_TEXT);
    Textbox->SetSel(0,4);
    Textbox->SetFocus();
    Textbox->ReplaceSel("BACK");
}

```

รูปที่ 2.66 ตัวอย่างการใช้ฟังก์ชัน ReplaceSel()

ให้ทดลองพิมพ์โปรแกรมลงในฟังก์ชัน OnOK() และ Build โปรแกรมเพื่อดูผลการทำงาน เมื่อรันโปรแกรม ให้พิมพ์ข้อความใดๆลงไป ในเท็กซ์บ็อกซ์ จากนั้นกดปุ่ม OK ฟังก์ชัน SetSel จะทำการเลือกตัวอักษรตั้งแต่ตำแหน่งที่ 0 ถึง 4 และฟังก์ชัน ReplaceSel ก็จะแทนที่ข้อความที่ถูกเลือกนี้ด้วย "BACK" ดังรูป ซึ่งได้พิมพ์คำว่า "CHAISITE" เมื่อกดปุ่ม OK คำว่า "CHAI" จะถูกแทนที่ด้วยคำว่า "BACK"



รูปที่ 2.67 ผลการรันการใช้ฟังก์ชัน ReplaceSel()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.67 (ต่อ) ผลการรันการใช้ฟังก์ชัน ReplaceSel()

ข้อความที่ถูกแทนที่ลงไป จะไม่สามารถ Undo กลับมาได้ เพราะพารามิเตอร์ตัวที่ 2 ของฟังก์ชัน ReplaceSel() มีค่าตั้งต้นเป็น FALSE หากต้องการให้สามารถ Undo กลับมาได้ก็จะต้องกำหนดค่าลงไป ดังรูปที่ 2.68

```
textbox->ReplaceSel("BACK",TRUE);
```

รูปที่ 2.68 การกำหนดให้ข้อความที่ถูกแทนที่สามารถ Undo ได้

2.10.36 การจำกัดจำนวนตัวอักษร

สามารถกำหนดจำนวนตัวอักษรที่สามารถพิมพ์ลงไปในแท็บซ์บ็อกซ์ได้ โดยใช้ฟังก์ชัน LimitText ซึ่งการเรียกฟังก์ชัน LimitText นี้ จะต้องเรียกมันในตอนเริ่มต้นโปรแกรม เพื่อเป็นการกำหนดค่าให้กับคอนโทรลก่อนการใช้งาน

จะนำโค้ดโปรแกรมที่ใช้ในการเรียกฟังก์ชัน LimitText ใส่ไว้ในฟังก์ชัน OnInitDialog() เพราะฟังก์ชันนี้ เป็นฟังก์ชันที่จะถูกเรียกเมื่อไดอะล็อกปรากฏบนจอภาพ มันจึงเป็นฟังก์ชันที่เหมาะสมที่สุดในการกำหนดค่าเริ่มต้นให้กับคอนโทรลในไดอะล็อก ดังโค้ดโปรแกรมต่อไปนี้ เป็นการกำหนดจำนวนตัวอักษรสูงสุดให้กับแท็บซ์บ็อกซ์โดยใช้ฟังก์ชัน LimitText

```

BOOL CsingleLineDlg::OnInitDialog(){
    CDialog::OnInitDialog();
    SetIcon(m_hIcon,TRUE);
    SetIcon(m_hIcon,FALSE);
    CEdit *textbox;
    textbox = (CEdit*)GetDlgItem(IDC_TEXT);
    textbox->LimitText(5);
    return TRUE;
}

```

รูปที่ 2.69 การจำกัดตัวอักษร

ให้เปิดไฟล์ CaiDlg.cpp จากแท็บ File View ในโปรเจกต์เวิร์กสเปซ และแก้ไขฟังก์ชัน OnInitDialog ให้เหมือนกับโค้ดโปรแกรมที่แสดงในข้างต้น ซึ่งเป็น โปรแกรมและทดลองรันเพื่อดู ผลการทำงาน จะพบว่าเมื่อพิมพ์ตัวอักษร ไปจนครบ 5 ตัวแล้ว จะพิมพ์ต่อไปไม่ได้อีก

2.10.37 การกำหนดข้อความเริ่มต้นให้กับเท็กซ์บ็อกซ์

สามารถกำหนดข้อความเริ่มต้นภายในเท็กซ์บ็อกซ์ได้ โดยฟังก์ชัน SetDlgItemText ซึ่งเป็น ฟังก์ชันหนึ่งในคลาส CWnd ที่ตรงข้ามกับฟังก์ชัน GetDlgItemText

การเขียน โค้ด โปรแกรมเพื่อกำหนดค่าเริ่มต้นให้กับคอนโทรลนี้จะต้องใส่เอาไว้ในฟังก์ชัน OnInitDialog () เช่นเดียวกับการเรียกฟังก์ชัน LimitTextดังรูปที่ 2.70

```

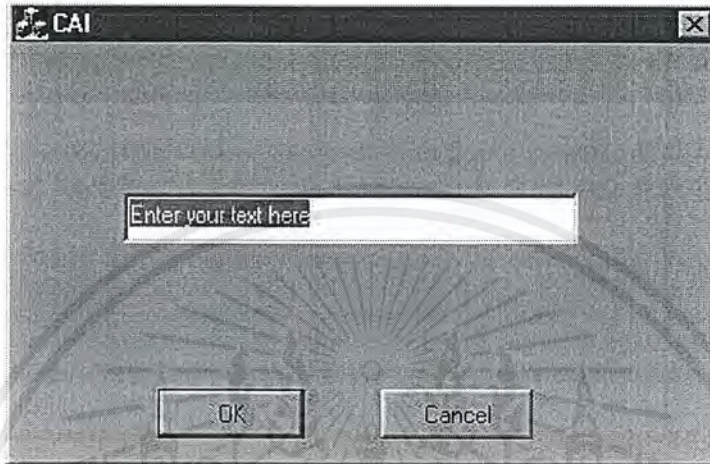
BOOL CsingleLineDlg::OnInitDialog(){
    Cdialog::OnInitDialog();
    SetIcon(m_hIcon,TRUE);
    SetIcon(m_hIcon,FALSE);
    SetDlgItemText(IDC_TEXT, "Enter your text here");
    Return TRUE;
}

```

รูปที่ 2.70 การกำหนดข้อความเริ่มต้นให้กับเท็กซ์บ็อกซ์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนักผู้เอาไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อรันโปรแกรม ฟังก์ชัน OnInitDialog จะถูกเรียกให้ทำงาน ข้อความ “Enter your text here” จะถูกแสดงลงในเท็กซ์บ็อกซ์ IDC_TEXT ทันทีดังรูปที่ 2.71



รูปที่ 2.71 ผลการรันการกำหนดค่าเริ่มต้น

ให้ลองสังเกตดูว่า เมื่อรันโปรแกรมแล้ว ไม่ได้เรียกฟังก์ชัน SetSel() เลยแต่ทำไมถึงมีแถบสีล้อมรอบข้อความที่อยู่ภายในเท็กซ์บ็อกซ์ นั่นเป็นเพราะว่าเท็กซ์บ็อกซ์นั้นได้ถูกจัดให้อยู่เป็นอันดับแรกใน Tab Order นั่นเอง เพราะฉะนั้น เมื่อไอเคอร์ถูกแสดงบนจอภาพโฟกัสจะจับอยู่ที่เท็กซ์บ็อกซ์ ก็เท่ากับว่าเป็นการเลือก (Select) คอนโทรลตัวนั้นไปในตัว จึงเห็นแถบสีล้อมรอบข้อความเมื่อโปรแกรมทำงาน

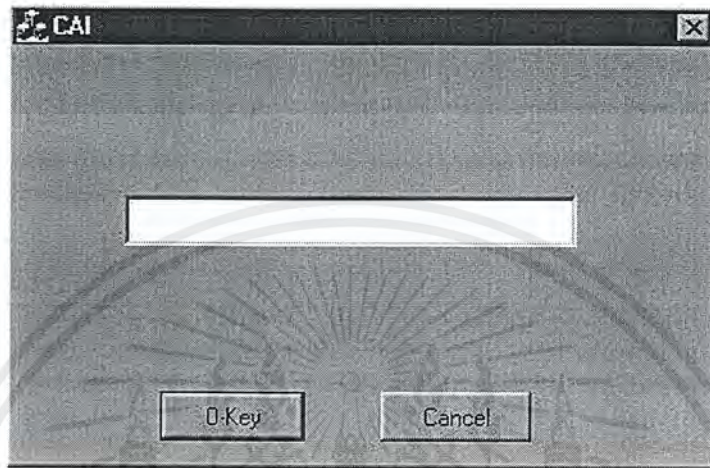
นอกจากจะสามารถกำหนดข้อความให้กับเท็กซ์บ็อกซ์ได้แล้ว ยังสามารถใช้ฟังก์ชัน SetDlgItemText และ GetDlgItemText กับคอนโทรลตัวอื่นได้อีก เช่นการเปลี่ยนข้อความแคปชั่น (Caption) บนปุ่มกด จากคำว่า OK ให้เป็น O-Key ก็สามารทำได้ โดยเขียนโค้ดโปรแกรมดังรูปที่ 2.72

```
SetDlgItemText (IDOK, "O-Key");
```

รูปที่ 2.72 การใช้ฟังก์ชัน SetDlgItemText() กับคอนโทรลอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ลองแก้ไขโค้ดโปรแกรม เพื่อความเข้าใจมากขึ้น ดังรูปที่ 2.73 แสดงผลการรันโปรแกรม เมื่อแก้ไขโค้ดข้างต้นแล้ว



รูปที่ 2.73 ผลการรันการเปลี่ยนแปลงข้อความกับคอนโทรลอื่น

2.10.38 การกำหนดให้เท็กซ์บ็อกซ์อยู่ในสถานะ ReadOnly

ในการกำหนดให้เท็กซ์บ็อกซ์เป็นแบบ Read-Only ผู้ใช้โปรแกรมจะไม่สามารถพิมพ์ข้อความใดๆลงไปในเท็กซ์บ็อกซ์ได้ แต่สามารถกำหนดข้อความภายในเท็กซ์บ็อกซ์ได้โดยตรงโดยใช้ฟังก์ชัน SetDlgItemText

การกำหนดให้เท็กซ์บ็อกซ์อยู่ในสถานะ Read-Only นั้น สามารถทำได้ 2 วิธีคือ

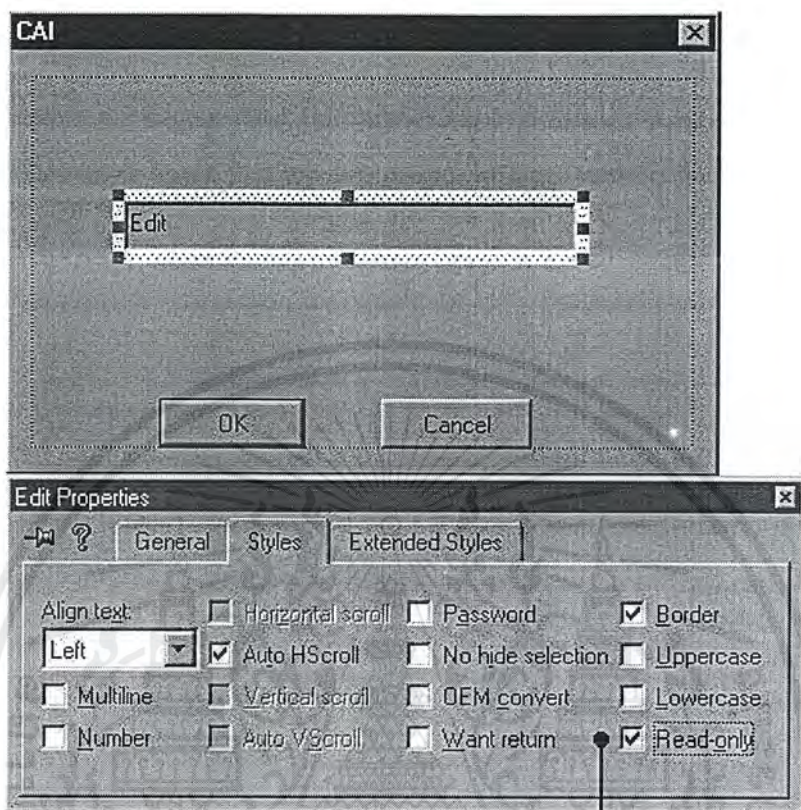
- 1) กำหนดค่าเริ่มต้นให้กับเท็กซ์บ็อกซ์เลย โดยใช้ไคอะลือกรีซอร์ส

ตัวอย่างในการกำหนดสถานะ Read-Only ให้กับเท็กซ์บ็อกซ์ ให้เรียกไคอะลือกรีซอร์ส โดยเปิดไคอะลือกรีซอร์ส IDD_SINGLELINE_DIALOG ของโปรเจกต์ SingleLine ในตัวอย่างข้างต้นขึ้นมา จากนั้นใช้การคลิกเมาส์ไปที่เท็กซ์บ็อกซ์ IDC_TEXT และกดปุ่ม <Alt+Enter> เพื่อเรียกไคอะลือกรีซอร์ส Edit Properties

จากไคอะลือกรีซอร์ส Edit Properties ให้เลือกไปที่แท็บ Styles และกาเครื่องหมาย Read-only เพียงเท่านั้น เท็กซ์บ็อกซ์ก็จะอยู่ในสถานะ Read-only แล้ว

เมื่อ build โปรเจกต์และทดลองรันโปรแกรม จะพบว่าไม่สามารถพิมพ์ข้อความใดๆลงในเท็กซ์บ็อกซ์ได้เลย แต่ยังสามารถกำหนดค่าต่างๆ ภายในเท็กซ์บ็อกซ์ได้เหมือนปกติ อาจจะใช้ฟังก์ชัน SetDlgItemText ดังที่ได้อธิบายไปในข้างต้น (สิ่งที่เปลี่ยนแปลงไปก็คือ ผู้ใช้จะไม่สามารถพิมพ์ข้อความใดๆลงไปในเท็กซ์บ็อกซ์ได้อีกแล้ว)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กาเครื่องหมายที่ Read only

รูปที่ 2.74 การกำหนดค่าเริ่มต้นให้อยู่ในสถานะอ่านอย่างเดียว

2) การเรียกฟังก์ชัน SetReadOnly เพื่อกำหนดสถานะ Read-Only ของเท็กซ์บ็อกซ์ ฟังก์ชัน SetReadOnly เป็นเมมเบอร์ฟังก์ชันของคลาส Cedit ใช้สำหรับกำหนดสถานะ Read-Only ให้กับเท็กซ์บ็อกซ์ ซึ่งสามารถเรียกใช้ได้เมื่อต้องการแต่การแก้ไขสถานะ Read-Only โดยใช้ไดอะล็อกอิดิตเตอร์ที่ได้เสนอไปข้างต้น จะเป็นการกำหนดสถานะในคอนเริ่มต้นให้กับ ไดอะล็อกโดยอัตโนมัติ

รูปแบบของฟังก์ชัน SetReadOnly มีดังนี้

```
BOOL SetReadOnly (BOOL bReadOnly =TRUE);
```

รูปที่ 2.75 รูปแบบของฟังก์ชัน SetReadOnly

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พารามิเตอร์ `bReadOnly` ถ้าเป็น `TRUE` หรือ `1` หมายถึง “จริง” ซึ่งก็คือกำหนดให้เท็กซ์บ็อกซ์อ่านได้อย่างเดียว แต่ถ้ากำหนดพารามิเตอร์ให้เป็น `FALSE` หรือ `0` เท็กซ์บ็อกซ์ก็จะกลับมาสู่สถานะที่สามารถรับข้อมูลจากผู้ใช้ได้เหมือนเดิม ซึ่งถ้าเรียกใช้ฟังก์ชัน `SetReadOnly` โดยไม่ต้องใส่พารามิเตอร์ให้กับมัน มันจะถือว่ากำหนดค่า `TRUE` โดยอัตโนมัติ

ตัวอย่างการเรียกใช้ฟังก์ชัน `SetReadOnly` มีดังรูปที่ 2.76

```
void CsingleLineDlg::OnOK(){
    CEdit *textbox;
    textbox = (CEdit*)GetDlgItem(IDC_TEXT);
    static BOOL ro=TRUE;
    if (ro==TRUE);
    else
        ro=!ro;
    textbox->SetReadOnly(ro);
}
```

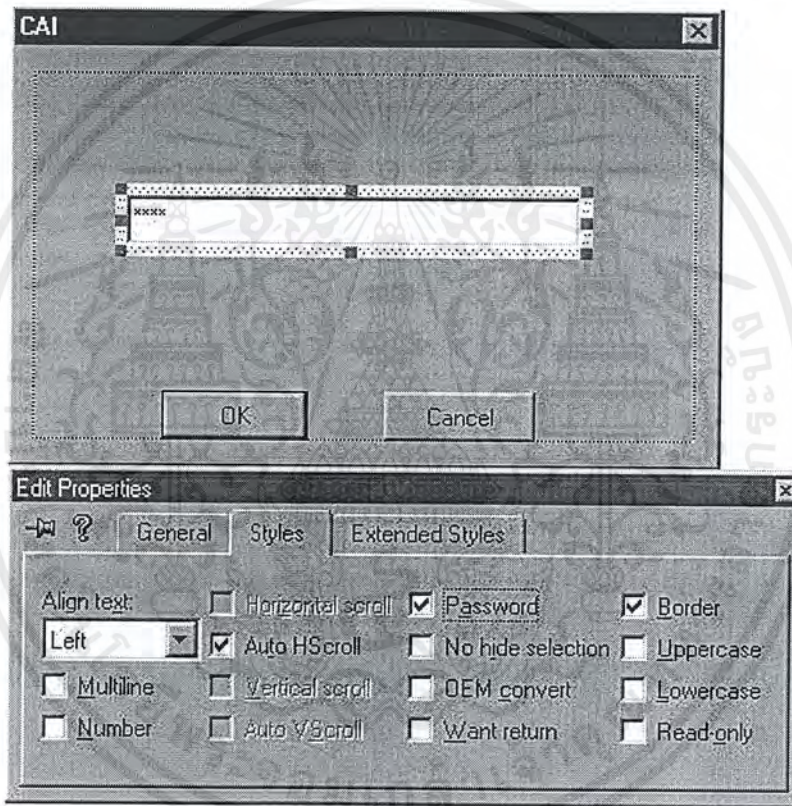
รูปที่ 2.76 ตัวอย่างการเรียกใช้ฟังก์ชัน `SetReadOnly`

จากฟังก์ชัน `OnOK` ในข้างต้น ให้ลองแก้ไขและทดลองรันโปรแกรมเพื่อดูผลการทำงานจะพบว่าเมื่อกดปุ่ม `OK` เท็กซ์บ็อกซ์จะสลับสถานะ `Read-Only` ไปๆมาๆ ในการกดปุ่ม `Ok` แต่ครั้งคราวนี้ให้มาทำความเข้าใจกับตัวโปรแกรมกันสักเล็กน้อย ให้สังเกตว่าในโปรแกรมนี้ได้กำหนดตัวแปร `ro` เป็นตัวแปรแบบ `BOOL` และเป็นตัวแปรแบบ `static` หมายความว่า ค่าของตัวแปร `ro` นี้จะไม่เปลี่ยนแปลงหรือสูญหายไป มันจะคงค่าเดิมอยู่ภายในเวลานั้น และภายในฟังก์ชันนั้นๆ จะเห็นได้ว่า ต้องกำหนดให้ตัวแปร `ro` เป็น `TRUE` ก่อน จากนั้นก็จะเข้าสู่เงื่อนไข `if` โดยจะตรวจสอบว่า `ro` เป็น `TRUE` หรือไม่ ถ้าเป็น `TRUE` ก็จะกำหนดให้ `ro` เปลี่ยนค่าไปเป็นตรงกันข้าม ซึ่งก็คือ `FALSE`

2.10.39 การสร้างเท็กซ์บ็อกซ์แบบ Password

คงเคยเห็นเท็กซ์บ็อกซ์ประเภทหนึ่งที่เวลาพิมพ์ข้อความใดๆลงไป มันจะกลายเป็นเครื่องหมายดอกจัน (*) เรียกเท็กซ์บ็อกซ์แบบนี้ว่า เท็กซ์บ็อกซ์แบบ Password

การกำหนดเท็กซ์บ็อกซ์ให้เป็นแบบ Password ทำได้โดยการแก้ไขโค้อะลือกริชอร์ส IDD_SINGLELINE_DIALOG โดยให้เรียกโค้อะลือกริชอร์ส Edit Properties ของเท็กซ์บ็อกซ์โดยกด <Alt+Enter> จากนั้นให้คลิกไปที่แท็บ Styles และกาเครื่องหมายในช่อง Password ดังรูปที่ 2.77



รูปที่ 2.77 การกำหนดเท็กซ์บ็อกซ์ให้เป็นแบบ Password โดยการแก้ไขโค้อะลือกริชอร์ส

จากนั้น ให้ลอง Build โปรเจกต์และรันโปรแกรมเพื่อดูผลการทำงานดังรูป แสดงผลการทำงานของโปรแกรม โดยเมื่อพิมพ์ข้อความใดๆลงในเท็กซ์บ็อกซ์ IDC_TEXT แล้ว จะพบว่าข้อความกลายเป็นเครื่องหมายดอกจันทั้งหมด

ในการอ่านข้อความที่เป็นเครื่องหมายดอกจันนั้น สามารถทำได้โดยใช้ฟังก์ชันที่เคยศึกษามาแล้ว นั่นก็คือ GetDlgItemText ถึงการแสดงผลของข้อความภายในเท็กซ์บ็อกซ์จะเป็นเครื่องหมายดอกจัน แต่ข้อความที่พิมพ์ลงไปก็จะเป็นข้อความเดิมไม่เปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ ยังสามารถเปลี่ยนการแสดงผลของเท็กชื่อบ็อกซ์แบบ Password จากเครื่องหมายดอกจัน(*) เป็นเครื่องหมายอื่นได้ โดยใช้ฟังก์ชัน SetPasswordChar ซึ่งมีรูปแบบดังรูปที่ 2.78

```
viod SetPasswordChar ( TCHAR ch);
```

รูปที่ 2.78 รูปแบบฟังก์ชัน SetPasswordChar

พารามิเตอร์ ch ก็คือค่ารหัสของอักขระที่จะได้เป็นเครื่องหมายของ Password นั้นเอง ปกติจะเป็นเครื่องหมายดอกจัน หากต้องการเปลี่ยนให้เป็นเครื่องหมายชาร์ป(#) หรือดอลลาร์(\$) ก็สามารทำได้โดยให้เพิ่มเติมโค้ดโปรแกรมลงในฟังก์ชัน OnInitDialog ภายในไฟล์ CaiDlg.cpp ให้เป็นดังรูปที่ 2.79

```
BOOL CsingleLineDlg::OnInitDialog();
Cdialog::OnInitDialog();
SetIcon(m_hIcon,TRUE);
SetIcon(m_hIcon,FALSE);
CEdit *textbox;
textbox = (CEdit*)GetDlgItem(IDC_TEXT);
textbox->SetPasswordChar('#');
return TRUE;
}
```

รูปที่ 2.79 การใช้ฟังก์ชัน SetPasswordChar

เมื่อ Build โปรแกรมและรันโปรแกรม จะพบว่า เมื่อพิมพ์ข้อความใดๆลงในเท็กชื่อบ็อกซ์ จะกลายเป็นเครื่องหมายชาร์ป(#)แทน

2.10.40 การใช้งานแท็กช็อบ็อกซ์แบบหลายบรรทัด

การสร้างแท็กช็อบ็อกซ์แบบหลายบรรทัดสามารถสร้างได้เช่นเดียวกับ แท็กช็อบ็อกซ์บรรทัดเดียวโดยนำแท็กช็อบ็อกซ์จากทูลบาร์คอนโทรลมาวางลงในไดอะล็อก และให้ปรับขนาดความสูงหลายบรรทัด โดยใช้ เมาส์ ลากเพื่อขยายขนาดของแท็กช็อบ็อกซ์ และกดปุ่ม <Alt+Enter> เพื่อเรียกไดอะล็อก Edit Properties ให้เปลี่ยนค่า ID ของคอนโทรลจาก IDC_EDIT1 ให้เป็น IDC_MESSAGE ดังรูปที่ 2.80



รูปที่ 2.80 การสร้างแท็กช็อบ็อกซ์แบบหลายบรรทัด

ภายใน Edit Properties ให้เลือกไปที่แท็บชื่อ Styles (แท็บที่ 2) จากนั้นให้กาเครื่องหมายที่ช่อง “MultiLine” และช่อง “Want Return” ดังรูปที่ 2.81

ปรับลำดับ Tab Order ของคอนโทรลภายในไดอะล็อก โดยกด <Ctrl+D> จากนั้นให้คลิกที่แท็กช็อบ็อกซ์ให้เป็นหมายเลข 1 และปุ่ม OK เป็นหมายเลข 2 และปุ่ม Cancel เป็นหมายเลข 3 ดังรูปที่ 2.82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับลำดับของ Tab Order นี้จะทำให้การโฟกัสของคอนโทรลมาโฟกัสอยู่ที่
 ฟิลด์บ็อกซ์เป็นครั้งแรกเมื่อรันโปรแกรมนี้

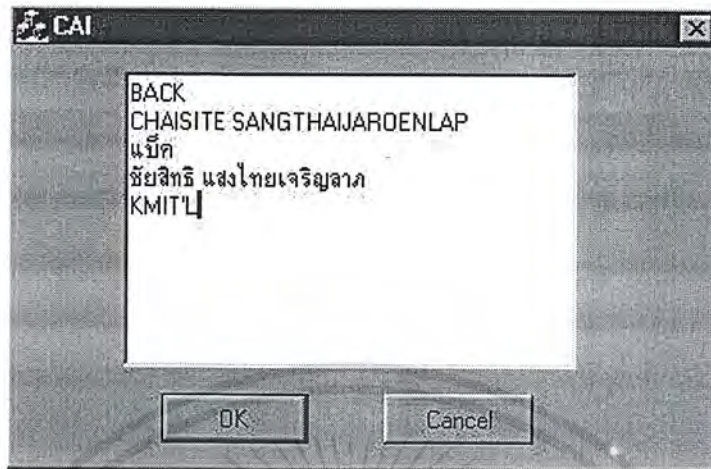


รูปที่ 2.81 การกำหนดรูปแบบให้ฟิลด์บ็อกซ์แบบหลายบรรทัด



รูปที่ 2.82 การปรับลำดับโฟกัสให้กับฟิลด์บ็อกซ์แบบหลายบรรทัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



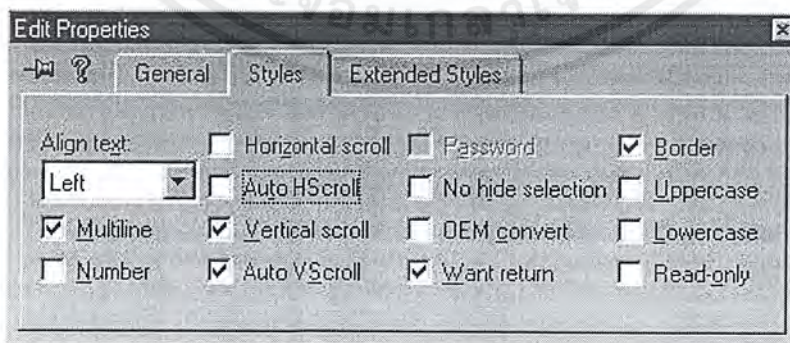
รูปที่ 2.83 ผลการรันโปรแกรม MultiLine

จากนั้นให้กด <F7> เพื่อทำการ Build โปรเจกต์และทดลองรันโปรแกรมโดยกด <Ctrl+F5> ดังรูปที่ 2.83 เป็นผลการรันโปรแกรม MultiLine

จากการรันโปรแกรม MultiLine ในข้างต้น จะเห็นได้ว่าแท็กชื่อบ็อกซ์ที่ได้สร้างไปนี้สามารถพิมพ์ข้อความลงไปได้หลายบรรทัด โดยสามารถกด <Enter> เพื่อขึ้นบรรทัดใหม่ได้ ซึ่งการที่สามารถกดปุ่ม Enter ขึ้นบรรทัดใหม่นี้ เป็นผลมาจากการกำหนดตัวเลือก “Want Return” นั่นเอง

2.10.41 ตัวเลือกในการสกอรัลและการตัดคำ

สามารถกำหนดให้แท็กชื่อบ็อกซ์มีความสามารถในการสกอรัลข้อความได้ โดยให้เรียนไดอะล็อก Edit Properties ของแท็กชื่อบ็อกซ์ขึ้นมา และเลือกไปที่แท็บ Styles ดังรูปที่ 2.84



รูปที่ 2.84 การทำให้แท็กชื่อบ็อกซ์มีความสามารถในการสกอรัลข้อความได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของตัวเลือกในการสกรอลล์ข้อความ มีให้เลือกดังนี้

- 1) **Horizontal Scroll** แสดงสกรอลล์บาร์แนวตั้งประกบแท็บบ็อกซ์
- 2) **Auto Hscroll** สกรอลล์ข้อความในแนวนอนแบบอัตโนมัติ
- 3) **Vertical Scroll** แสดงสกรอลล์บาร์แนวนอนประกบแท็บบ็อกซ์
- 4) **Auto Vscroll** สกรอลล์ข้อความในแนวตั้งแบบอัตโนมัติ

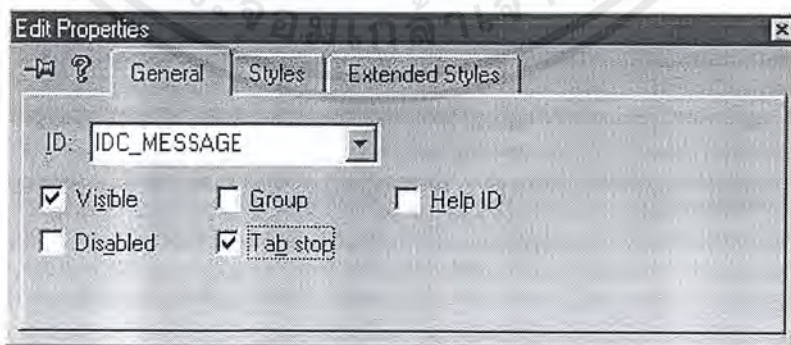
หากต้องการให้แท็บบ็อกซ์แบบหลายบรรทัดมีความสามารถในการตัดคำอัตโนมัติ จะต้องเอาเครื่องหมายที่อยู่ในช่อง Auto Hscroll ออกไป เพราะเมื่อพิมพ์ข้อความไปเรื่อยๆจนเกินขอบเขตของคอนโทรล มันก็จะตัดคำให้อัตโนมัติ และก็ควรเอาเครื่องหมายในช่อง Auto Vscroll และ Vertical Scroll เอาไว้ด้วย เพราะเมื่อพิมพ์ข้อความไปเรื่อยๆ ในแนวตั้ง เมื่อสุดขอบด้านล่างมันก็จะเลื่อน บรรทัดใหม่ขึ้นมาให้พิมพ์ต่อไปเรื่อยๆและสามารถใช้สกรอลล์บาร์เลือกเพื่อไปยังบรรทัดที่ต้องการได้

2.10.42 ตัวเลือกอื่นๆ ของแท็บบ็อกซ์

สามารถกำหนดคุณสมบัติอื่นๆ ให้กับแท็บบ็อกซ์ได้อีก นอกเหนือจากที่ได้กล่าวมาในข้างต้น โดยจะใช้ไคอะล๊อคอิติเตอร์ในการแก้ไข ซึ่งสามารถใช้ได้กับแท็บบ็อกซ์ทั้งแบบบรรทัดเดียว และแบบหลายบรรทัดการยกเลิกลำดับ Tab Order

1) การยกเลิกลำดับ Tab Order

สามารถยกเลิกลำดับ Tab Order ของคอนโทรลตัวใดๆก็ได้ภายในไคอะล๊อคนั้นคือ เมื่อกดปุ่ม <Tab> คอนโทรลที่ถูกยกเลิก Tab Order นี้จะไม่ได้รับโฟกัส สามารถทำได้โดยนำเครื่องหมายที่อยู่หน้าช่อง Tab Stop ในแท็บ General ออกไปดังรูปที่ 2.85



รูปที่ 2.85 การยกเลิกลำดับ Tab Order

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) การกำหนด Align Text

สามารถกำหนดให้ข้อความที่พิมพ์ลงไปในเท็กซ์บ็อกซ์นั้นชิดซ้ายหรือชิดขวา หรือว่าอยู่กึ่งกลางได้ โดยเลือกตัวเลือกในช่อง Align Text ในแท็บ Styles

3) การแสดงตัวอักษรแบบ Lower-Case และ Upper-Case

สามารถกำหนดให้สิ่งที่พิมพ์ลงไปในเท็กซ์บ็อกซ์นั้นเป็นตัวเล็กหรือตัวใหญ่อย่างเดียวได้ โดยกาเครื่องหมายที่ช่อง Lowercase หรือช่อง Uppercase ซึ่งสามารถเลือกได้เพียงแค่ช่องเดียว

2.10.43 แมสเสจของเท็กซ์บ็อกซ์

จากที่ได้ศึกษาเรื่องของปุ่มกดมานั้น จะเห็นได้ว่าแมสเสจที่เกิดจากปุ่มกดสามารถมีได้ 2 แมสเสจนั่นคือ BN_CLICKED และ BN_DOUBLECLICKED แมสเสจทั้งสองตัวนี้เป็นแมสเสจที่เกิดจากการตอบสนองหรือตอบโต้กับผู้ใช้ ซึ่งในการใช้งานเท็กซ์บ็อกซ์ ก็จะมีแมสเสจในลักษณะนี้เช่นเดียวกัน

แมสเสจของเท็กซ์บ็อกซ์ที่สามารถควบคุมได้มีดังนี้

- 1) EN_CHANGE แมสเสจนี้จะเกิดขึ้น เมื่อมีการเปลี่ยนแปลงข้อความในเท็กซ์บ็อกซ์ (EN_UPDATE ก็เช่นกัน)
- 2) EN_SETFOCUS แมสเสจนี้จะเกิดขึ้น เมื่อมีการโฟกัสที่เท็กซ์บ็อกซ์ (ตัวเคอร์เซอร์จะพริบอยู่ที่เท็กซ์บ็อกซ์)
- 3) EN_KILLFOCUS แมสเสจนี้จะเกิดขึ้น เมื่อมีการย้ายโฟกัสไปที่คอนโทรลตัวอื่น
- 4) EN_MAXTEXT แมสเสจนี้จะเกิดขึ้นเมื่อไม่สามารถพิมพ์ข้อความใดๆลงไปได้แล้ว
- 5) EN_ERRSPACE แมสเสจนี้จะเกิดขึ้น เมื่อเท็กซ์บ็อกซ์อยู่ในสภาวะ Out of Memory
- 6) EN_HSCROLL แมสเสจนี้เกิดขึ้นจากผู้ใช้ทำการเลื่อนสกรอลบาร์ในแนวนอน
- 7) EN_VSCROLL แมสเสจนี้เกิดขึ้นจากผู้ใช้ทำการเลื่อนสกรอลบาร์ในแนวตั้ง

2.10.44 การใช้ DDX จัดการข้อมูลในไดอะล็อก

DDX ย่อมาจาก Dialog Data Exchange นั่นคือ กระบวนการแลกเปลี่ยนข้อมูลภายในไดอะล็อก ระหว่างคอนโทรลต่างๆภายในไดอะล็อกกับตัวแปรที่เป็นสมาชิกภายในคลาส

DDX เป็นกระบวนการหนึ่งของ MFC ที่จะช่วยให้จัดการข้อมูลในไดอะล็อกได้ง่ายมากยิ่งขึ้น จะยกตัวอย่างเพื่อให้เข้าใจในการใช้ DDX จัดการข้อมูล

ไดอะล็อกที่สร้างขึ้นเท็กซ์บ็อกซ์นั้น การที่จะนำข้อมูลที่อยู่ในเท็กซ์บ็อกซ์มาใช้จะต้องเรียกฟังก์ชัน GetDlgItemText เพื่อนำข้อมูลที่อยู่ในเท็กซ์บ็อกซ์มาเก็บลงตัวแปรดังโค้ดโปรแกรมต่อไปนี้

```
char data [50];
GetDlgItemText(IDC_TEXT,data,50);
```

รูปที่ 2.86 การนำข้อมูลในเท็กซ์บ็อกซ์มาเก็บลงตัวแปร

โค้ดโปรแกรมในรูปที่ 2.86 ฟังก์ชัน GetDlgItemText จะเอาข้อความที่ผู้ใช้พิมพ์ลงไป ในเท็กซ์บ็อกซ์ IDC_TEXT มาเก็บไว้ที่ตัวแปร data ทันที เมื่อสิ้นสุดบรรทัดนี้ และก็จะสามารถนำเอาตัวแปร data ไปใช้งานได้ หรือในอีกกรณีหนึ่ง ถ้าหากต้องการกำหนดให้เท็กซ์บ็อกซ์หรือคอนโทรลสแตติกแสดงข้อความใดๆออกมา ก็จะต้องเรียกฟังก์ชัน SetDlgItemText เพื่อกำหนดข้อความที่ต้องการให้แสดงที่คอนโทรลตัวนั้น ดังโค้ดโปรแกรมในรูปที่ 2.87

```
SetDlgItemText(IDC_TEXT,"This is a test");
```

รูปที่ 2.87 การกำหนดข้อความที่ต้องการให้แสดงที่คอนโทรล

จากบรรทัดในข้างต้น ข้อความว่า "This is a test" ก็จะถูกนำไปแสดงในคอนโทรล IDC_TEXT โดยอัตโนมัติ การใช้ฟังก์ชัน GetDlgItemText และ SetDlgItemText ทั้งสองตัวนี้เป็นกระบวนการจัดการข้อมูลแบบธรรมดาที่ได้ศึกษามาแล้ว ซึ่งจะแตกต่างจากการใช้ DDX ที่ศึกษากันนี้สักเล็กน้อย เพราะในการใช้ DDX เข้ามาช่วย ไม่จำเป็นต้องเรียกฟังก์ชัน GetDlgItemText หรือ SetDlgItemText แต่จะเรียกฟังก์ชันที่ชื่อ UpdateData เพียงแค่ฟังก์ชันเดียวเท่านั้น

ฟังก์ชัน UpdateData เป็นฟังก์ชันที่ใช้ในกระบวนการ DDX ฟังก์ชันนี้จะทำหน้าที่ อัปเดตข้อมูลระหว่างคอนโทรลและตัวแปร ฟังก์ชัน UpdateData นี้ต้องการพารามิเตอร์หนึ่งตัว ซึ่งเป็นแบบ Boolean (TRUE กับ FALSE) รูปแบบของฟังก์ชัน UpdateData ที่แสดงไว้ในเอกสารอ้างอิงของ MFC มีดังรูปที่ 2.88

```
BOOL UpdateData(BOOL bSaveAndValidate=TRUE);
```

รูปที่ 2.88 รูปแบบของฟังก์ชัน UpdateData

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พารามิเตอร์ `bSaveAndValidate` จะใช้สำหรับกำหนดทิศทางของการอัปเดตข้อมูล ซึ่งมีค่าคือ `FALSE` (ถ้าหากเรียกฟังก์ชันนี้โดยไม่ใส่พารามิเตอร์ มันจะเป็น `TRUE` โดยอัตโนมัติ) ความแตกต่างของการกำหนดค่า `TRUE` กับ `FALSE` ให้กับฟังก์ชันตัวนี้ก็คือ

ถ้าเรียกฟังก์ชัน `UpdateData` โดยใส่พารามิเตอร์ `TRUE` ให้กับฟังก์ชัน ข้อมูลที่อยู่ภายในคอนโทรลจะถูกนำมาเก็บลงตัวแปร (Member Variable) ที่กำหนดโดยอัตโนมัติ

ถ้าใส่พารามิเตอร์ `FALSE` ให้กับฟังก์ชันค่าที่อยู่ในตัวแปรจะถูกแสดงออกทางคอนโทรล นี่คือการแตกต่างของการใส่ค่า `TRUE` และ `FALSE` ให้กับฟังก์ชัน `UpdateData` จะมาสร้างโปรเจกต์ทดสอบการใช้งาน DDX กับเพื่อที่จะได้ทำความเข้าใจยิ่งขึ้น โดยให้สร้างโปรเจกต์ใหม่ขึ้นมาโดยใช้ AppWizard ให้สร้างโปรเจกต์แบบ Dialog-based ดังวิธีที่ได้ศึกษามาแล้ว โดยกำหนดชื่อโปรเจกต์ให้เป็น DDX

เมื่อ AppWizard สร้างซอร์สโค้ดต้นแบบและโปรเจกต์เวิร์กสเปซเสร็จเรียบร้อยแล้ว ให้เปิดไดอะล็อกกริซอร์ส `IDD_DDX_DIALOG` ขึ้นมาแก้ไขโดยให้ปรับแต่งไดอะล็อกดังต่อไปนี้

ปรับขนาดของไดอะล็อก และปรับตำแหน่งของปุ่มกด

แก้ไขสแตติกคอนโทรล `IDC_STATIC` จาก “TODO:Place dialog controls here” ให้เป็นคำว่า “Enter Text”

เพิ่มแท็บที่ล็อกเข้าไปในไดอะล็อก และกำหนดให้ค่า ID ของแท็บที่ล็อกนี้เป็น `IDC_DDXTEXT`

คราวนี้ก็ได้โปรเจกต์ DDX ที่แก้ไขไดอะล็อกกริซอร์สเสร็จเรียบร้อยแล้ว ต่อไปจะมาแก้ไขที่ซอร์สโค้ดโปรแกรมกัน โดยจะนำเอา DDX เข้ามาใช้ในการจัดการข้อมูลในแท็บที่ล็อกแทนการใช้ฟังก์ชัน `GetDlgItemText` โดยให้ทำตามขั้นตอนต่อไปนี้

1) ขั้นตอนที่ 1 เรียก ClassWizard

ให้กดปุ่ม `<ctrl+W>` เพื่อเรียก MFC ClassWizard ขึ้นมา จากนั้นให้คลิกไปที่แท็บ Member Variables

ในแท็บ Member Variables จะเป็นแท็บที่ใช้สำหรับเพิ่มตัวแปรที่เป็นสมาชิกของคลาส (Member Variables) เพื่อใช้ในกระบวนการ DDX

2) ขั้นตอนที่ 2 เพิ่มตัวแปรให้กับคอนโทรล `IDC_DDXTEXT`

ขั้นตอนนี้ จะทำการเพิ่มตัวแปรที่จะใช้สำหรับเก็บค่าของข้อความในแท็บที่ล็อก โดยให้คลิกในช่อง Control Ids ไปที่ `IDC_DDXTEXT` จากนั้นคลิกที่ปุ่ม Add Variable จะพบกับไดอะล็อก Add Member Variable

จากไดอะล็อก Add Member Variable จะพบว่าในช่อง Member variable name มีคำว่า “m_” ปรากฏอยู่และเคอร์เซอร์ก็รอรับอยู่ด้วยในช่องนี้ จะเป็นช่องที่ให้ใส่ชื่อตัวแปรที่ต้องการลงไป โดยจะมี “m_” นำหน้า (ถ้าไม่ต้องการให้มี “m_” นำหน้าตัวแปร ก็สามารถลบออกได้) ให้พิมพ์ชื่อตัวแปรลงไปให้เป็น m_ddxtext ดังรูปที่ 2.89

ในช่อง Category ให้เลือกไปที่ Value เพราะจะทำการสร้างตัวแปรตัวใหม่ และในช่อง Variable Type ให้เลือกไปที่ CString (ซึ่งกำหนดเอาไว้ให้แล้วโดยอัตโนมัติ) ซึ่ง CString นี้เป็นตัวแปรประเภทหนึ่งที่มีลักษณะการทำงานคล้ายๆกับตัวแปรชุด char[...]นั่นเอง แต่จะต่างกันตรงที่ CString จะเป็นคลาสที่ทำหน้าที่จัดการข้อมูลสตริง

เมื่อเพิ่มตัวแปร m_ddxtext เรียบร้อยแล้ว จะพบว่า MFC ClassWizard ได้ทำการเพิ่มตัวแปรให้กับ ดังรูปที่ 2.89 ให้กดปุ่ม OK เพื่อปิดไดอะล็อก MFC ClassWizard

3) ขั้นตอนที่ 3 แมปเมสเสจให้กับปุ่ม OK

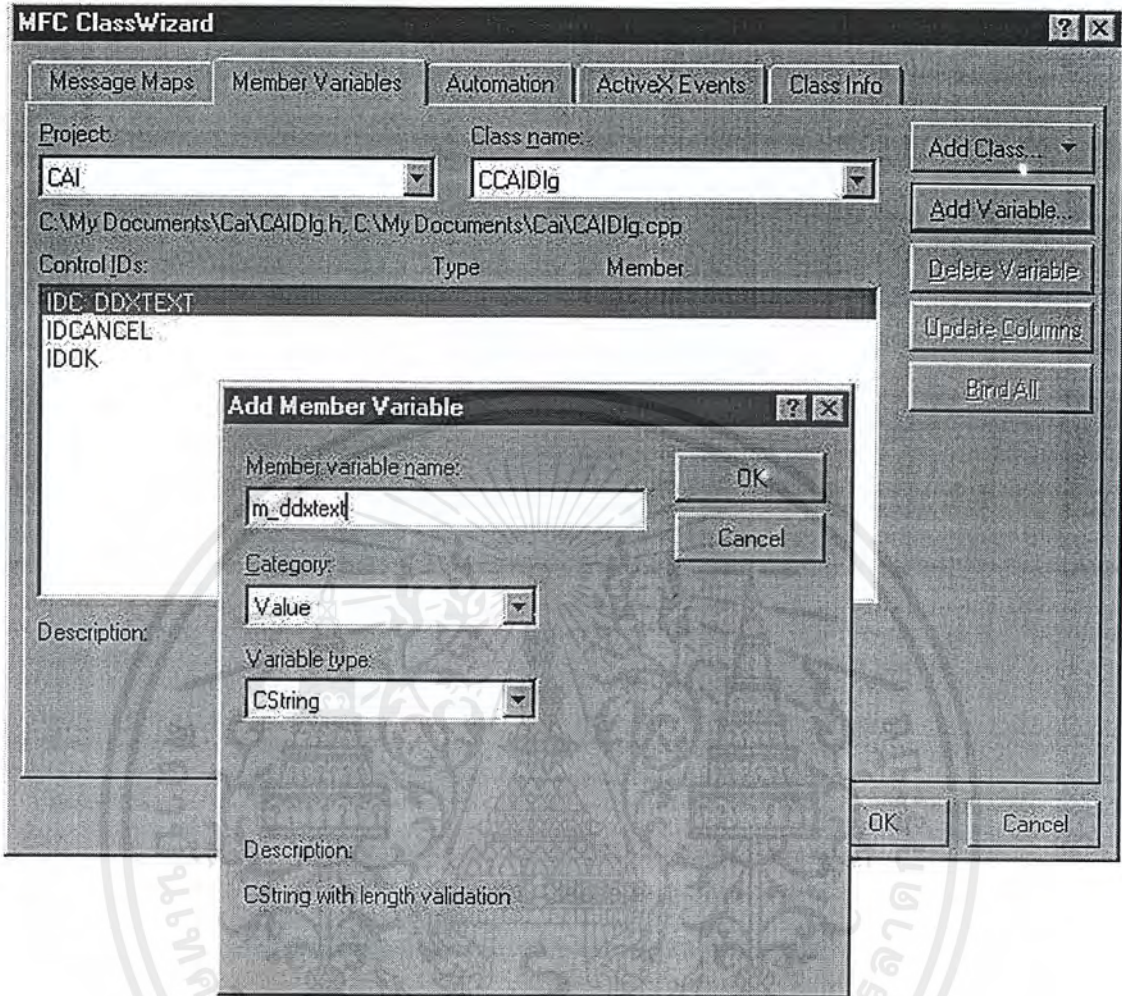
ให้เรียก ClassWizard อีกครั้ง โดยกด<ctrl+W> และเลือกไปที่แท็บ MessageMaps จากนั้น ให้แมปปุ่ม OK กับเมสเสจ BN_CLICKED โดยให้เพิ่มเมมเบอร์ฟังก์ชัน OnOKลงไป

4) ขั้นตอนที่ 4 แก้ไขโค้ดโปรแกรมให้กับฟังก์ชัน OnOK

เมื่อแมปเมสเสจและเพิ่มเมมเบอร์ฟังก์ชัน OnOK ในขั้นตอนที่ 3 เรียบร้อยแล้ว ให้คลิกที่ปุ่ม Edit Code บนไดอะล็อก MFCClassWizard เพื่อแก้ไขฟังก์ชัน OnOK ดังรูปที่ 2.90

5) ขั้นตอนที่ 5 Build โปรเจกต์และทดลองรันโปรแกรม

ให้กด <F7> เพื่อ Build โปรเจกต์ และกด <ctrl+F5 >เพื่อรันโปรแกรม จะพบกับไดอะล็อก DDX และแท็บบ็อกซ์ ให้พิมพ์ข้อความที่พิมพ์ออกมาดังรูปที่ 2.91



รูปที่ 2.89 การเพิ่มตัวแปรให้กับคอนโทรล

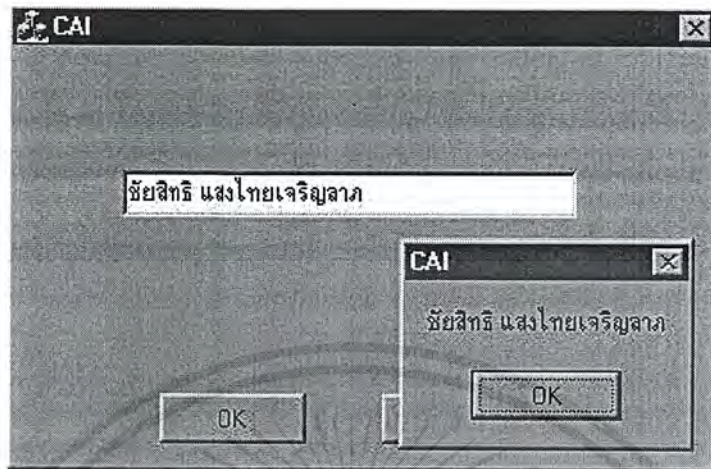
```

Void CDDXDlg::OnOK(){
    UpdateData(TRUE);
    MessageBox(m_ddxtext);
}

```

รูปที่ 2.90 ตัวอย่างการใช้ตัวแปรชนิด DDX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.91 ผลการรัน โปรแกรมที่ใช้ตัวแปรชนิด DDX

จากโปรเจกต์ DDX ที่สร้างไปนั้น ได้ใช้กระบวนการ DDX เข้ามาช่วยในการจัดการข้อมูลให้กับคอนโทรลในไดอะล็อก จะเห็นได้ว่าในฟังก์ชัน OnOK ได้เรียกฟังก์ชัน UpdateData โดยใส่พารามิเตอร์ TRUE ลงไป

เมื่อสิ้นสุดบรรทัดนี้ ข้อความที่ปรากฏอยู่ในเท็กซ์บ็อกซ์ IDC_DDXTEXT ก็จะถูกนำมาเก็บไว้ที่ตัวแปร m_ddxtext โดยอัตโนมัติ นั่นเป็นเพราะว่า ได้กำหนดให้ตัวแปร m_ddxtext เป็นตัวแปรที่ใช้ในกระบวนการ DDX โดยทำงานร่วมกับคอนโทรล IDC_DDXTEXT นั้นเอง

จากนั้นจะเรียกฟังก์ชัน MessageBox เพื่อแสดงข้อความในตัวแปร m_ddxtext ออกทางจอภาพ จะเห็นได้ว่าไม่ได้เรียกฟังก์ชัน GetDlgItemText เพื่อนำข้อมูลมาเก็บไว้ที่ตัวแปรเลย เรียกเพียงแค่ฟังก์ชัน UpdateData(TRUE) เท่านั้น นี่คือประโยชน์ของการใช้ DDX อย่างหนึ่ง

เมื่อนำเอาข้อความในเท็กซ์บ็อกซ์มาเก็บลงในตัวแปร m_ddxtext ได้ โดยเรียกฟังก์ชัน UpdateData(TRUE) ก็สามารถใส่ฟังก์ชัน UpdateData(FALSE) เพื่อนำข้อมูลที่อยู่ในตัวแปร m_ddxtext ไปแสดงในเท็กซ์บ็อกซ์โดยใช้ฟังก์ชัน UpdateData(FALSE) ได้

ให้ลองเปลี่ยนโค้ด โปรแกรมในฟังก์ชัน OnOK ให้เป็นดังรูปที่ 2.92

```

Void CDDXDlg::OnOK(){
M_ddxtext = "KMIT'L KMIT'L KMIT'L";
UpdateData(FALSE);
}

```

รูปที่ 2.92 ตัวอย่างการใช้ฟังก์ชัน UpdateData(FALSE)

จะเห็นได้ว่า m_ddxtext ซึ่งเป็นตัวแปรแบบ CString ที่ได้สร้างขึ้นมาโดยใช้ Class Wizard นั้น ได้ถูกกำหนดให้เก็บข้อความว่า "KMIT'L KMIT'L KMIT'L" เอาไว้ จากนั้นเมื่อทำการเรียก ฟังก์ชัน UpdateData(FALSE) ข้อความที่อยู่ในตัวแปร m_ddxtext ก็จะถูกนำไปแสดงในเท็กซ์บ็อกซ์

ถ้าต้องการนำข้อมูลมาเก็บลงในตัวแปร จะต้องเรียกฟังก์ชัน UpdateData(TRUE) ก่อนเป็นอันดับแรก จึงจะเอาตัวแปรไปใช้งานได้ แต่ถ้าต้องการให้ค่าที่อยู่ในตัวแปรไปแสดงในคอนโทรล ก็จะต้องกำหนดค่าลงในตัวแปรก่อน จึงจะเรียกฟังก์ชัน UpdateData(FALSE)

2.10.45 Check box และ Radio box

เช็คบ็อกซ์ (Check box) และเรดิโอบ็อกซ์ (Radio box) เป็นคอนโทรลประเภทหนึ่งที่ใช้สำหรับกำหนดค่าจริง (True) หรือไม่จริง (False) จะพบการใช้คอนโทรลเช็คบ็อกซ์และเรดิโอบ็อกซ์ ในลักษณะของงานที่ใช้เป็นตัวเลือก (Option) ในโปรแกรมต่างๆไป ซึ่งทำให้โปรแกรมมีความยืดหยุ่นและมีประสิทธิภาพมากขึ้น

2.10.46 การสร้างและใช้งานเช็คบ็อกซ์

เช็คบ็อกซ์ (Check box) เป็นคอนโทรลที่มีรูปแบบการใช้งานที่ง่ายตัวหนึ่ง นั่นก็เพราะว่าเช็คบ็อกซ์สามารถมีได้เพียง 2 สถานะเท่านั้น คือ

- 1) จริง (TRUE) มีการกาเครื่องหมายที่เช็คบ็อกซ์
- 2) 'ไม่จริง (FALSE) 'ไม่มีการกาเครื่องหมายที่เช็คบ็อกซ์

เช็คบ็อกซ์จะสามารถทำงานได้โดยตัวมันเพียงตัวเดียว เพราะมันมีการเปลี่ยนแปลงค่าภายในตัวมันเองเท่านั้น ซึ่งจะต่างจากเรดิโอบ็อกซ์ ซึ่งจะต้องอยู่กันเป็นกลุ่มตั้งแต่ 2 ตัว ขึ้นไป สำหรับการสร้างสามารถสร้างได้ตามขั้นตอนต่อไปนี้

- 1) ขั้นตอนที่ 1 แก้ไขไดอะล็อกบ็อกซ์

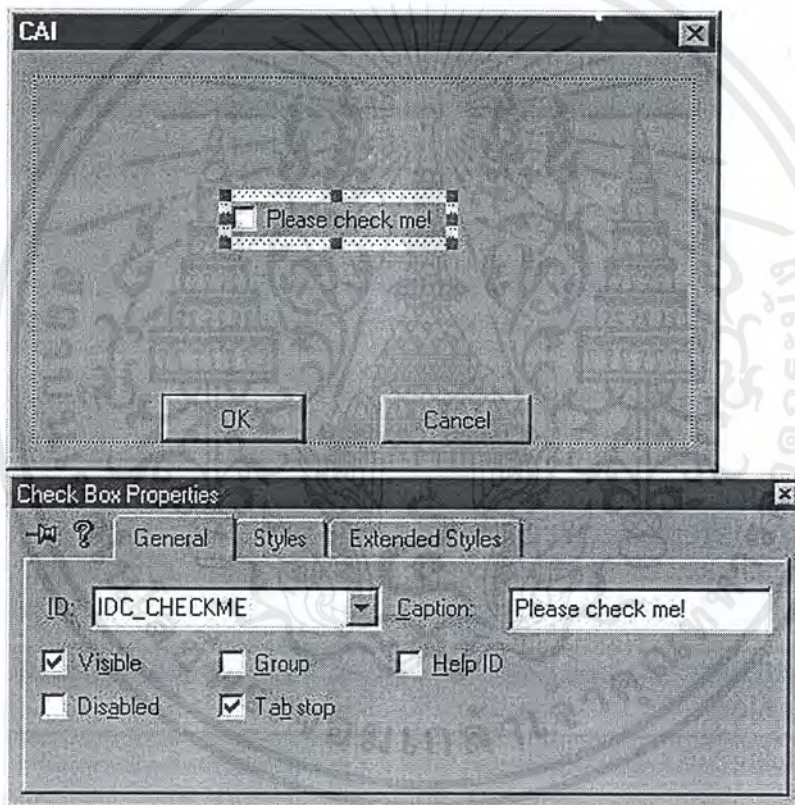
ให้เลือกแท็บ Resource View ของโปรเจกต์เวิร์กสเปซ Check และ ดับเบิลคลิกไดอะล็อก

บ็อกซ์ IDD_CHECK_DIALOG โดยปรับขนาดของไดอะล็อกและย้ายปุ่ม OK กับ Cancel มาอยู่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางด้านล่างของไดอะล็อก และลบสแตติคคอนโทรลที่เขียนว่า “TODO:Place dialog control here.” ออกไป

2) ขั้นตอนที่ 2 สร้างเช็คบ็อกซ์

คราวนี้ให้นำเช็คบ็อกซ์จากคอนโทรลทูลบาร์มาวางในไดอะล็อก โดยวางไว้ตรงกลางของไดอะล็อก จากนั้นกดปุ่ม <Alt+Enter> หรือกดปุ่ม <Enter> เพื่อเรียกไดอะล็อก Check Box Properties จากนั้นจะกำหนดให้เช็คบ็อกซ์นี้มีค่า ID เป็น IDC_CHECKME และกำหนดให้มีแคปชั่น (Caption) แสดงคำว่า “Please check me!” ดังรูปที่ 2.93

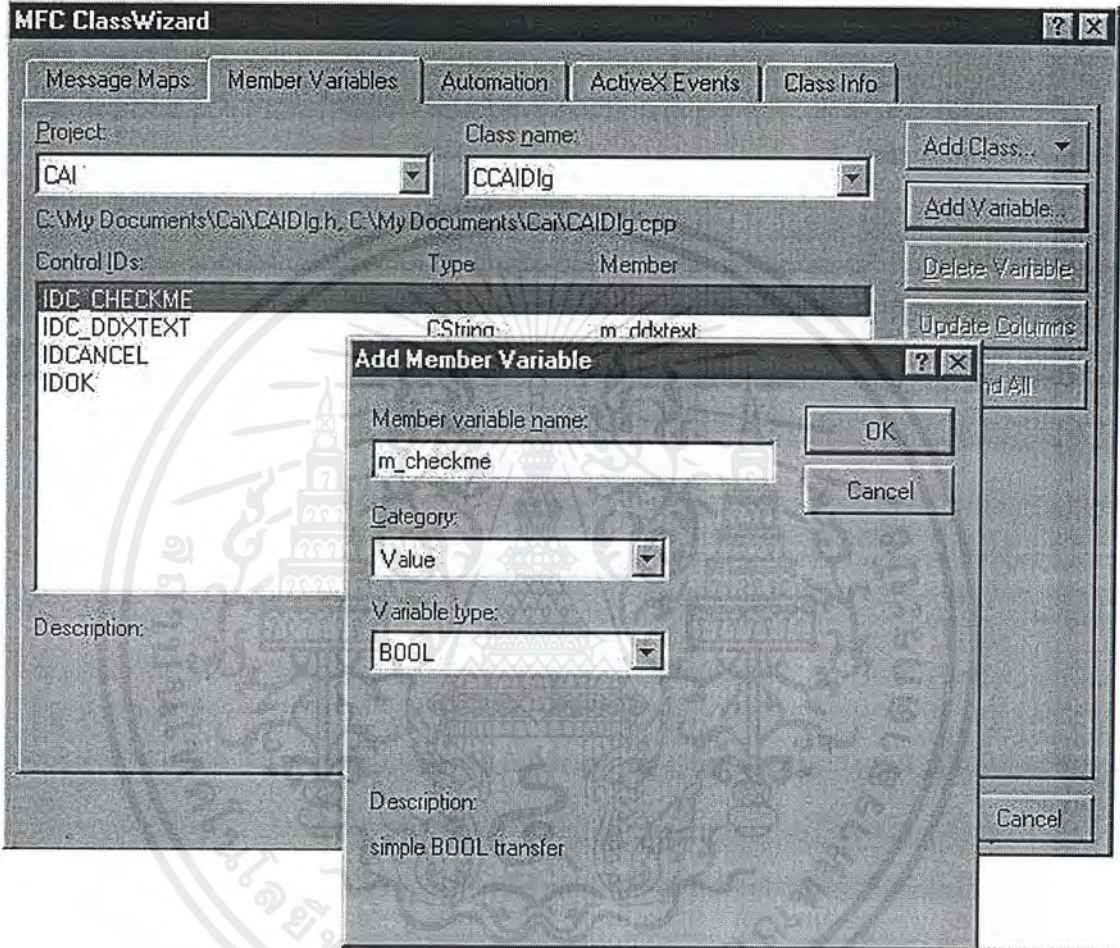


รูปที่ 2.93 การสร้างเช็คบ็อกซ์

3) ขั้นตอนที่ 3 สร้างตัวแปร m_checkme

ขั้นตอนต่อไปนี้ก็คือนการสร้างตัวแปร m_checkme (หรืออาจจะใช้ชื่ออื่นที่ต้องการก็ได้) เพื่อใช้ในกระบวนการ DDX เพื่อให้ง่ายต่อการเขียนโค้ดโปรแกรม เพราะจะใช้ตัวแปรตัวนี้เป็นตัวบอกสถานะของเช็คบ็อกซ์ว่ามีการกาเครื่องหมายอยู่หรือไม่

การสร้างตัวแปรตัวนี้ ทำได้โดยใช้ ClassWizard โดยให้กด <Ctrl+W> เพื่อเรียก MFC ClassWizard ขึ้นมา จากนั้นให้คลิกไปที่แท็บ Member Variables ดังรูปที่ 2.94



รูปที่ 2.94 การสร้างตัวแปร DDX ให้กับเช็คบ็อกซ์

4) ขั้นตอนที่ 4 แมปเมสเสจ BN_CLICKED ให้กับปุ่ม OK

คราวนี้จะมาทำการแมปเมสเสจ BN_CLICKED ให้กับปุ่ม OK กัน โดยจุดประสงค์ก็คือจะใช้ปุ่ม OK นี้เป็นปุ่มทดสอบเช็คบ็อกซ์ โดยเมื่อมีการกดปุ่ม OK เมื่อใด ให้มีเมสเสจบ็อกซ์แสดงสถานะของเช็คบ็อกซ์ ปรากฏขึ้น ถ้าเช็คบ็อกซ์ถูกกาเครื่องหมาย ให้แสดงข้อความออกมาว่า "Checked" และถ้าเช็คบ็อกซ์ว่างเปล่าไม่มีการกาเครื่องหมาย ให้แสดงข้อความออกมาว่า "No Check!!"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) ขั้นตอนที่ 5 แก้ไขฟังก์ชัน OnOK

เมื่อเพิ่มฟังก์ชัน OnOK เรียบร้อยแล้ว ให้ทำการแก้ไขฟังก์ชัน โดย กดปุ่ม Edit Code บน ไอคอน MFC ClassWizard ก็จะพบกับหน้าต่างแสดงรายละเอียดของไฟล์ CaiDlg.cpp และ ฟังก์ชัน CCaiDlg::OnOK ให้แก้ไขฟังก์ชันให้เป็นดังรูปที่ 2.95

```
Void CCaiDlg::OnOK{
UpdateData(TRUE);
if (m_checkme==TRUE)
MessageBox("check");
else
MessageBox("no check");
}
```

รูปที่ 2.95 การแก้ไขฟังก์ชันให้ให้กับ โปรแกรมเช็คบ็อกซ์

จากนั้นให้ทดลอง Build โปรแกรมโดยกดปุ่ม <F7> จากนั้นกดปุ่ม <ctrl+F5> เพื่อทำการรัน โปรแกรม จะพบกับไอคอน Check ดังรูป ให้ลองกดปุ่ม Ok ดู จะพบว่า เมื่อเช็คบ็อกซ์ "Plesase check me!" ยังไม่ถูกกาเครื่องหมาย มันจะมีเมสเสจบ็อกซ์ แสดงข้อความว่า "No check" แต่ถ้าลอง กาเครื่องหมายลงในเช็คบ็อกซ์ และกดปุ่ม OK ก็พบว่า เมสเสจบ็อกซ์จะแสดงคำว่า "Check"

2.10.47 การสร้างและใช้งานเรดิโอบ็อกซ์

เรดิโอบ็อกซ์ (Radio Box) เป็นคอนโทรลอีกตัวหนึ่งซึ่งจะใช้ในกรณีที่ต้องการคำตอบจากผู้ ใช้ หรือเป็นตัวเลือกให้กับผู้ใช้โปรแกรม เช่นเดียวกับเช็คบ็อกซ์ แต่เรดิโอบ็อกซ์จะต่างจาก เช็คบ็อกซ์ตรงที่เรดิโอบ็อกซ์จะต้องทำงานร่วมกัน ตั้งแต่ 2 ตัวขึ้นไป นั่นหมายความว่า เรดิโอบ็อกซ์ จะต้องมากกว่า 2 ตัว มันถึงจะทำงานได้ เพราะการเลือกของเรดิโอบ็อกซ์เป็นการเลือกตัวเลือกใด ตัวเลือกหนึ่งเหมือนกับการเลือกคำตอบจากข้อสอบแบบ Choice

เพราะฉะนั้น ถ้าสร้างเรดิโอบ็อกซ์ขึ้นมาเพียงตัวเดียวในไอคอนก็จะมีเพียงตัวเลือกเดียว ไม่มีตัวเลือกอื่นๆให้เลือกนั้นจึงไม่ใช่วิธีการสร้างเรดิโอบ็อกซ์ที่ถูกตั้ง จะต้องจำวิธีการใช้งานของ เรดิโอบ็อกซ์ให้ดี เพราะมีกฎเกณฑ์เกี่ยวกับการใช้งานเล็กน้อย ซึ่งจะได้พูดถึงกันต่อไป

1) ขั้นตอนที่ 1 สร้างโปรเจกต์ Radio

มาทดลองสร้างโปรเจกต์ใหม่ อีกครั้งเพื่อทดสอบการทำงานของเรดิโอบ็อกซ์กัน โดยให้สร้างโปรเจกต์แบบ dialog-based โดยใช้ AppWizard

2) ขั้นตอนที่ 2 แก้ไขไดอะล็อก

ให้เปิดไดอะล็อกริซอร์ส IDD_RADIO_DIALOG จากนั้นให้เอาปุ่ม OK กับ Cancel วางไว้ด้านล่างของไดอะล็อก และนำเรดิโอบ็อกซ์จากทูลบาร์มาวางไว้ในไดอะล็อก

เรดิโอบ็อกซ์ที่นำมาวางทั้ง 2 ตัว ไดอะล็อกอิดิตเตอร์จะกำหนดค่า ID ให้เป็น IDC_RADIO1 และ IDC_RADIO2 ตามลำดับ ให้แก้ไขค่า ID กับแคปชั่นให้เป็นอย่างนี้

เรดิโอบ็อกซ์ IDC_RADIO1 เปลี่ยนค่า ID ให้เป็น IDC_YESBUTTON และเปลี่ยนแคปชั่นเป็น YES

เรดิโอบ็อกซ์ IDC_RADIO2 เปลี่ยนค่า ID ให้เป็น IDC_NOBUTTON และเปลี่ยนแคปชั่นเป็น NO

3) ขั้นตอนที่ 3 แมปเมสเสจให้กับปุ่ม OK

ทำการแมปเมสเสจให้กับปุ่ม OK เพื่อที่จะใช้ในการทดสอบเรดิโอบ็อกซ์โดยกดปุ่ม <Ctrl+W> เพื่อเรียก MFC ClassWizard จากนั้นแมปเมสเสจ BN_CLICKED ให้กับปุ่ม IDOK ซึ่ง ClassWizard จะเพิ่มฟังก์ชัน OnOK ลงไปให้ ดังที่ได้ทดลองสร้างกันมาแล้ว ให้แก้ไขโปรแกรมดังรูปที่ 2.96

```

Void CcaiDlg::OnOK() {
    CButton *yesbutton,*nobutton;
    yesbutton = (CButton*)GetDlgItem(IDC_YESBUTTON);
    nobutton = (CButton*)GetDlgItem(IDC_NOBUTTON);
    if((yesbutton->GetCheck())==1)
        MessageBox("YES");
    if((nobutton->GetCheck())==1)
        MessageBox("NO");
}

```

รูปที่ 2.96 โปรแกรมทดสอบการทำงานของ เรดิโอบ็อกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ทดลองคอมไพล์โปรแกรมและรันโปรแกรมเพื่อดูผลการทำงาน ก็จะพบว่า ถ้าเลือกตัวเลือก YES และกดปุ่ม OK เมสเสจบ็อกซ์ก็จะแสดงคำว่า “YES” และถ้าเลือกไปที่ตัวเลือก NO และกดปุ่ม OK เมสเสจบ็อกซ์ก็จะแสดงว่า “NO” ซึ่งก็จะสามารถเลือกได้เพียงตัวเลือกเดียวเท่านั้น ซึ่งก็คือไม่ YES ก็ NO

เรดิโอบ็อกซ์ที่มีมากกว่า 2 ตัวขึ้นไป สามารถจัดกลุ่มหรือจัดพวกให้กับเรดิโอบ็อกซ์ได้ เพราะถ้าไม่จัดกลุ่มให้กับมัน มันจะถือว่าเรดิโอบ็อกซ์ทุกตัวในไดอะล็อกเป็นพวกเดียวกัน จะมาลองแก้ไขโปรแกรม radio โดยเพิ่มเรดิโอบ็อกซ์ลงไป ในไดอะล็อกอีก 3 ตัว โดยให้เปิด ไดอะล็อกรีซอร์ส IDD_RADIO_DIALOG และนำเรดิโอบ็อกซ์มาวางลงในไดอะล็อกเพิ่มอีก 3 ตัว โดยกำหนดให้แคปชั่นเป็น Option 1, Option 2 และ Option 3 และกำหนด id ให้เป็น IDC_OPTION1, IDC_OPTION2 และ IDC_OPTION3 ตามลำดับ

จะเห็นได้ว่า เรดิโอบ็อกซ์ทั้ง 5 ตัวนั้น แบ่งออกเป็น 2 กลุ่มด้วยกันคือ

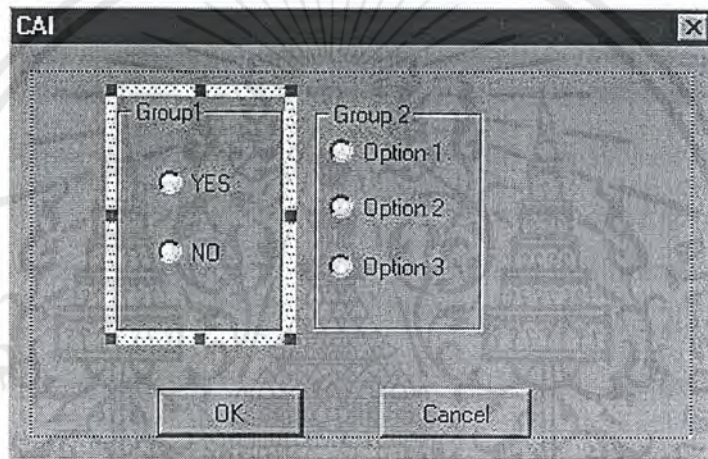
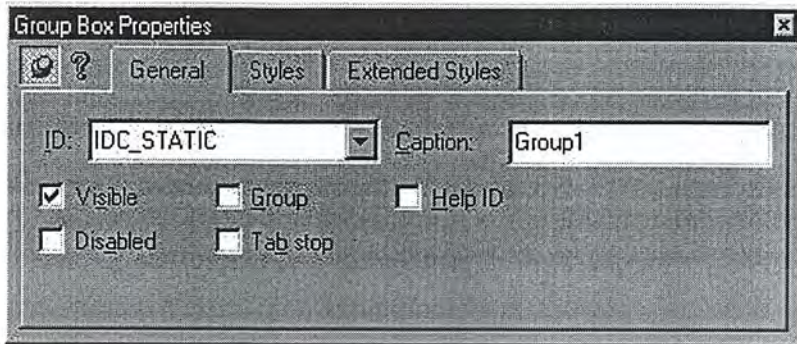
กลุ่มที่ 1 ประกอบด้วยเรดิโอบ็อกซ์ YES กับ NO

กลุ่มที่ 2 ประกอบด้วยเรดิโอบ็อกซ์ Option 1, Option 2 และ Option 3

ให้ลองทดสอบไดอะล็อก โดยกดปุ่ม <ctrl+T> จะพบว่าเมื่อเลือกตัวเลือกเรดิโอบ็อกซ์ในไดอะล็อก จะสามารถเลือกได้เพียงอย่างเดียวเท่านั้น หมายความว่า มันจะอยู่กันคนละกลุ่ม กล่าวคือในกลุ่มที่ 1 จะต้องมีการสลับค่ากันแค่ YES กับ NO เท่านั้น และกลุ่มที่ 2 ก็ควรจะสลับกันอยู่แค่ Option 1, Option 2 และ Option 3 เท่านั้น

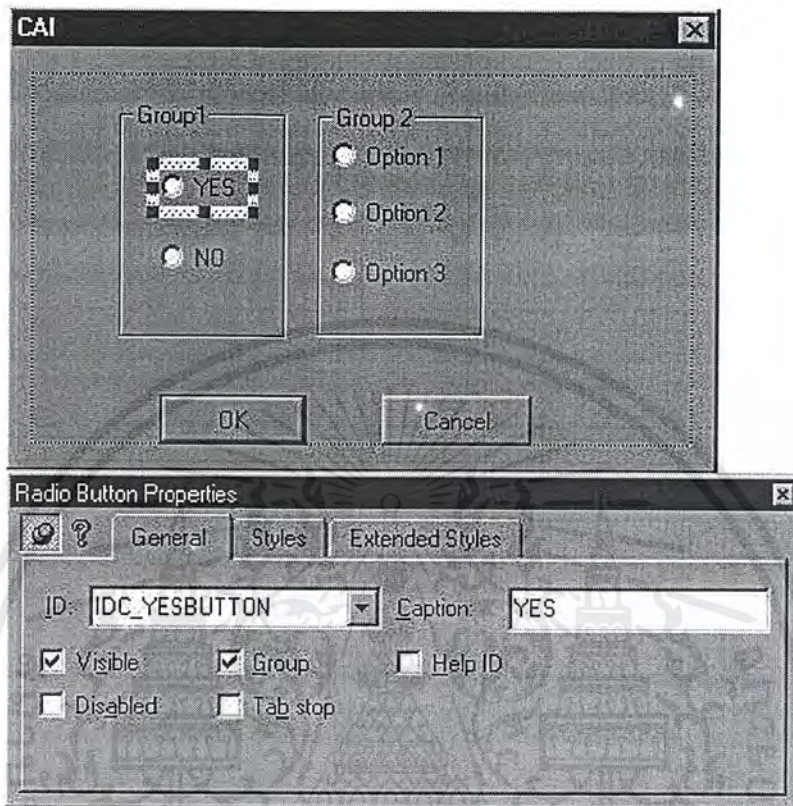
สามารถจับกลุ่มของเรดิโอบ็อกซ์ได้โดยใช้กรุปบ็อกซ์ (group box) เข้ามาช่วย นั่นคือจะต้องเอากรุปบ็อกซ์ 2 อันมาครอบเรดิโอบ็อกซ์เอาไว้ เพื่อแยกให้มันอยู่กันคนละกลุ่ม

ให้นำกรุปบ็อกซ์มาวางในไดอะล็อก โดยใช้วิธีการวางเหมือนกับการวาดรูปสี่เหลี่ยมรอบปุ่ม yes กับปุ่ม no มีแคปชั่นเป็น “Group 1” และกรุปบ็อกซ์ที่ล้อมรอบปุ่ม option1, option2 และ option3 นั้นมีแคปชั่นเป็น “Group 2” ดังรูปที่ 2.97



รูปที่ 2.97 การวางกรุปคอนโทรล

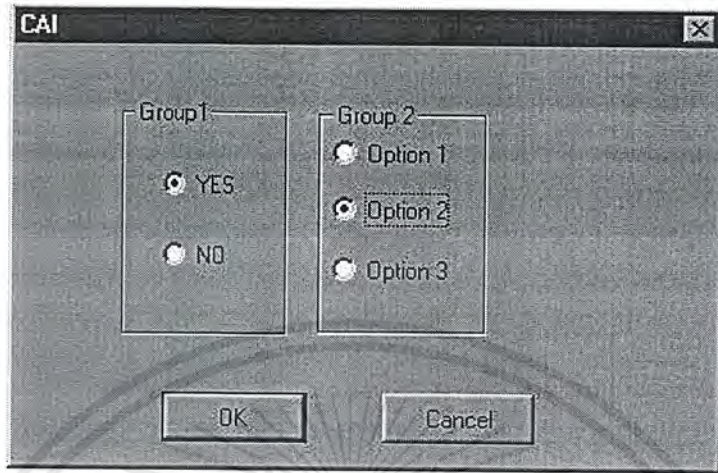
จากนั้นก็ถึงขั้นตอนที่สำคัญ นั่นคือ จะต้องกำหนดให้ตัวเลือกภายในกรอบเป็นกลุ่มเดียวกัน โดยจะทำให้เรดิโอบ็อกซ์ YES กับ NO เป็นพวกเดียวกัน โดยให้เรียกไดอะล็อก properties ของตัวเรดิโอบ็อกซ์ YES ขึ้นมาก่อน โดยคลิกเมาส์ที่ปุ่ม YES และกด <ALT+ENTER> จากนั้นให้ทำเครื่องหมายที่ช่อง Group ดังรูปที่ 2.98



รูปที่ 2.98 การจัดกลุ่มให้กับเรดิโอบ็อกซ์

ต่อไปก็กำหนดตัวเลือกให้กับกลุ่ม Group 2 โดยให้เรียกไดอะล็อก Properties ของเรดิโอบ็อกซ์ตัวบนสุด ซึ่งก็คือ Option1 ขึ้นมา จากนั้นให้กาเครื่องหมายที่ช่อง Group

ให้กดปุ่ม <Ctrl+T>เพื่อทดสอบไดอะล็อก จะเห็นได้ว่าคราวนี้เรดิโอบ็อกซ์ทั้ง 2 กลุ่มจะทำงานกันโดยไม่มายุ่งเกี่ยวกับอีก กลุ่มใครกลุ่มมัน นั่นเป็นเพราะว่าได้จัดแบ่งกลุ่มของตัวเลือกเอาไว้ให้แล้ว ดังรูปที่ 2.99



รูปที่ 2.99 ผลการจัดกลุ่มให้กับเรดิโอบ็อกซ์

ให้ลองเลือกตัวเลือกในกลุ่ม Group 1 และเลือกตัวเลือกในกลุ่ม Group 2 ถ้าต้องการสร้างเรดิโอบ็อกซ์หลายๆกลุ่ม จะต้องแบ่งกลุ่มให้กับเรดิโอบ็อกซ์ด้วยคอนโทรลที่ชื่อกรุปบ็อกซ์ ดังตัวอย่างโปรเจกต์ Radio ที่ได้ทดลองสร้างไปนั่นเอง

2.10.48 การสร้างและควบคุมลิสต์บ็อกซ์

การสร้างและการใช้งานลิสต์บ็อกซ์ก็เหมือนกับการใช้งานคอนโทรลตัวอื่นๆแต่ตัวเลือกของลิสต์บ็อกซ์จะมีมากกว่า ให้ดำเนินการตามขั้นตอนต่อไปนี้ เพื่อสร้างและทดสอบการทำงานของลิสต์บ็อกซ์

- 1) ขั้นตอนที่ 1 สร้างโปรเจกต์ใหม่ list box
- 2) ขั้นตอนที่ 2 แก้ไขไดอะล็อกรีซอร์ส

จะมาแก้ไขกันที่ไดอะล็อกรีซอร์สของโปรแกรม โดยปรับแต่งขนาดของไดอะล็อกและลบสแตติกคอนโทรล TODO ออก และย้ายตำแหน่งของปุ่ม OK กับ Cancel มาอยู่ทางด้านล่าง จากนั้นก็นำคอนโทรล Listbox จากทูลบาร์คอนโทรลมาวางในไดอะล็อก

เปลี่ยนค่า ID ของลิสต์บ็อกซ์ จาก IDC_LIST1 ให้เป็น IDC_MYLISTBOX โดยให้กด <ALT+ENTER> เพื่อเรียกไดอะล็อก List Box Properties ขึ้นมา

- 3) ขั้นตอนที่ 3 สร้างออบเจกต์ของลิสต์บ็อกซ์

ทำการสร้างออบเจกต์ของลิสต์บ็อกซ์ IDC_MYLISTBOX เพื่อใช้ในการเพิ่มข้อมูลลงในรายการ ซึ่งจะต้องใช้ออบเจกต์ของลิสต์บ็อกซ์ในการจัดการเรื่องนี้ให้กดปุ่ม <Ctrl+W> เพื่อเรียก

MFC ClassWizard จากนั้นเลือกไปที่แท็บที่ 2 Member Variables ให้เลือกค่า ID ของลิสต์บ็อกซ์ที่เอกสารนี้เป็นเอกสารที่สแกนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนงานวิชาการ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อ IDC_MYLISTBOX จากนั้นกดปุ่ม Add Variable.. และเพิ่มตัวแปรใหม่ที่ชื่อ m_cmylistbox ลงไป โดยกำหนดค่าในช่อง Category เป็น Control

จากนั้นให้เลือกไปที่แท็บ Message Maps และดูที่ช่อง Member function ทางด้านล่างของไคอะล็อก ให้ดับเบิลคลิกที่ตัวเลือกฟังก์ชัน OnInitDialog เพื่อแก้ไขโค้ดโปรแกรมของฟังก์ชันนี้

4) ขั้นตอนที่ 4 แก้ไขฟังก์ชัน OnInitDialog

ขั้นตอนนี้ จะเขียนโค้ดโปรแกรมเพื่อเพิ่มข้อความลงในช่องของลิสต์บ็อกซ์ เพื่อที่ว่า เมื่อรันโปรแกรมเมื่อไรก็จะมีข้อความอยู่ในลิสต์บ็อกซ์ จะใช้ออบเจกต์ของลิสต์บ็อกซ์ m_cmylistbox นี้เรียกฟังก์ชัน AddString ซึ่งเป็นฟังก์ชันที่ใช้ในการเพิ่มข้อความลงในลิสต์บ็อกซ์ รูปแบบของฟังก์ชัน AddString นั้นมีดังรูปที่ 2.100

```
int AddString (LPCTSTR lpszitem)
```

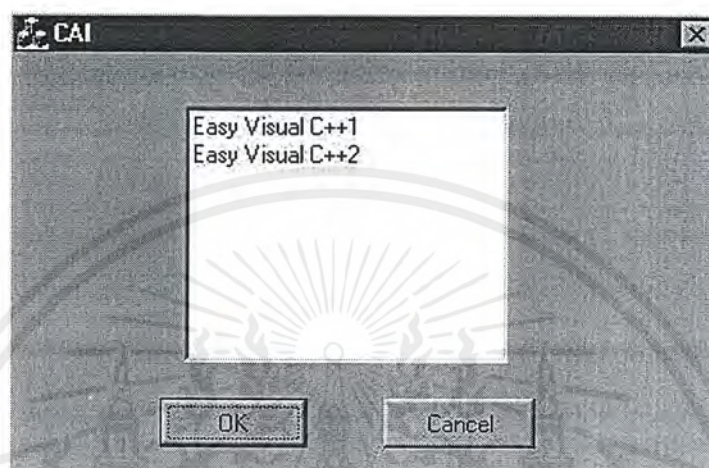
รูปที่ 2.100 รูปแบบการใช้ฟังก์ชัน AddString

เพิ่มโค้ดโปรแกรมลงไปในฟังก์ชัน On/initDialog ดังรูปที่ 2.101

```
BOOL CCaiDlg::OnInitDialog(){
Cdialog ::OnInitDialog();
SetIcon(m_hIcon,TRUE);
SetIcon(m_hIcon,FALSE);
m_cmylistbox.AddString("Easy Visual C++1");
m_cmylistbox.AddString("Easy Visual C++2");
return TRUE ;
}
```

รูปที่ 2.101 โปรแกรมทดสอบการทำงานของลิสต์บ็อกซ์

เมื่อแก้ไขฟังก์ชัน OnInitDialog ดังโค้ดโปรแกรม ในข้างต้นเสร็จแล้ว ให้กดปุ่ม <F7> เพื่อ Build โปรเจกต์ และกด <Ctrl +F5>เพื่อรันโปรแกรมจะพบกับไดอะล็อก และลิสต์บ็อกซ์ซึ่งมีข้อความว่า Easy Visual C++1 และ Easy Visual C++2 แสดงอยู่ ดังรูปที่ 2.102



รูปที่ 2.102 ผลการทดสอบ โปรแกรมลิสต์บ็อกซ์

ถ้าต้องการเพิ่มข้อความเข้าไปในลิสต์บ็อกซ์ทีละเยอะๆเป็น 10 หรือ 20 ข้อความ จะต้องนำเอาเทคนิคของการวนลูปเข้ามาช่วย นั่นคือจะใช้ลูป for ทำการวนลูปในจำนวนที่ต้องการเพื่อเพิ่มข้อความลงในช่องรายการ ดังโค้ดโปรแกรมหาดังรูปที่ 2.103

```

        BOOL CCaiDlg::OnInitDialog(){
        CDialog ::OnInitDialog();
        SetIcon(m_hIcon,TRUE);
        SetIcon(m_hIcon,FALSE);
        for (int i=1;i<=10;i++){
        m_mylistbox.AddString("Easy Visual C++");
        }
        return TRUE;
        }
    
```

รูปที่ 2.103 การนำการวนรอบมาใช้กับลิสต์บ็อกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโค้ดโปรแกรมที่เพิ่มเข้าไป จะเห็นได้ว่าลูป for จะวนตั้งแต่ เลข 1 จนถึง 10 ไปเรื่อยๆ และเรียกฟังก์ชัน AddString เพื่อเพิ่มข้อความลงในลิสต์บ็อกซ์จนครบ 10 ครั้ง

นอกจากนี้ ยังสามารถคิดแปลง หรือจัดฟอร์แมตของข้อความที่ต้องการได้ก่อนที่จะเพิ่มลงไป ในลิสต์บ็อกซ์ได้ ตัวอย่างเช่น หากต้องการให้ภายในลิสต์บ็อกซ์แสดงข้อความว่า

Easy Visual C++1

Easy Visual C++2

Easy Visual C++3

.....

Easy Visual C++20

สามารถทำได้โดยใช้ฟังก์ชัน sprintf ในการจัดการฟอร์แมตของข้อความ ดัง โค้ด โปรแกรม ในรูปที่ 2.104

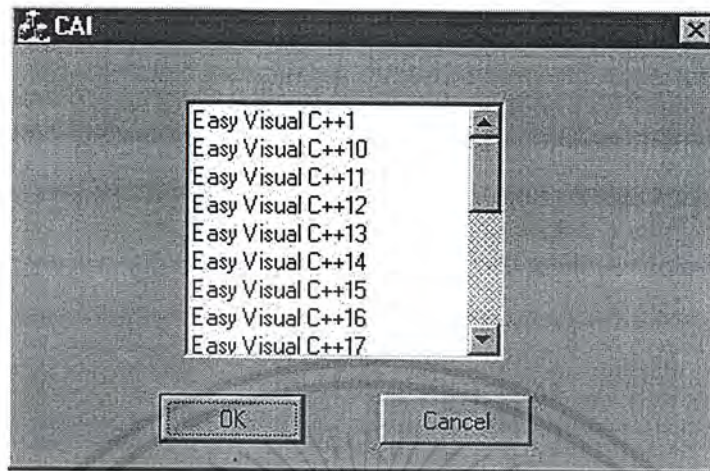
```

BOOL CcaiDlg::OnInitDialog(){
    Cdialog ::OnInitDialog();
    SetIcon(m_hIcon,TRUE);
        SetIcon(m_hIcon,FALSE);
    char str [20];
    for (int i=1;i<=20;i++){
        sprintf(str,"Easy Visual C++%d",i);
        m_cmylistbox.AddString(str);
    }
    return TRUE;
}

```

รูปที่ 2.104 โปรแกรมการวนรอบที่ปรับปรุงใหม่

เมื่อทำการ Build และทดลองรัน โปรแกรม จะได้ผลการรันโปรแกรมดังรูปที่ 2.105



รูปที่ 2.105 ผลการรัน โปรแกรมลิสต์บ็อกซ์ที่ปรับปรุงใหม่

จากโค้ดโปรแกรมที่ได้เพิ่มเข้าไปนั้น กำหนดตัวแปร str ให้เป็นแบบ char มีขนาด 20 ตัวอักษร จากนั้นฟังก์ชัน sprintf จะจัดฟอร์แมตของสตริง str โดยให้ %d แปรเปลี่ยนไปตามจำนวนครั้งของลูป for และจึงทำการเพิ่มลงไปในลิสต์บ็อกซ์

อีกสิ่งหนึ่งที่จะสังเกตได้จากผลการรัน โปรแกรม จะเห็นว่าข้อความที่อยู่ภายในลิสต์บ็อกซ์รายการนั้น ได้เรียงกันตามลำดับเลข ซึ่งทำให้ดูไม่สวยและไม่เป็นลำดับ เพราะฉะนั้นจะต้องแก้ไขตรงจุดนี้ โดยการบอกยกเลิก การ Sort ของลิสต์บ็อกซ์

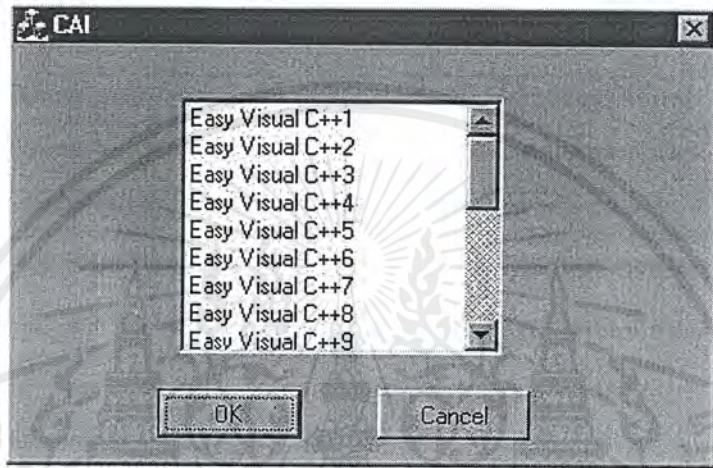
การยกเลิกการ Sort ของลิสต์บ็อกซ์ ทำได้โดยเปิดไดอะล็อกบ็อกซ์ IDD_LISTBOX_DIALOG ขึ้นและแก้ไข Properties ของลิสต์บ็อกซ์ โดยเลือกไปที่แท็บ Styles และยกเลิกเครื่องหมายในช่อง Sort ออกไป ดังรูปที่ 2.106



รูปที่ 2.106 การยกเลิกการ Sort ของลิสต์บ็อกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นให้ Build โปรเจกต์อีกครั้งและทดลอง รันโปรแกรมเพื่อดูผลการทำงาน ดังรูปที่ 2.101 จะเห็นว่า การเรียงลำดับของลิสต์บ็อกซ์จะเปลี่ยนใหม่ โดยจะเรียงตามลำดับของการเรียก ฟังก์ชัน AddString นั้นเอง



รูปที่ 2.107 ผลการรันเมื่อนำเครื่องหมาย Sort ของลิสต์บ็อกซ์ออก

2.10.49 คลาสที่ใช้จัดการกับลิสต์บ็อกซ์

คลาส Clistbox เป็นคลาสที่ใช้ในการจัดการลิสต์บ็อกซ์ เป็นคลาสหนึ่งภายใน MFC ภายในคลาสนี้ได้มีฟังก์ชันมากมายที่ใช้สำหรับจัดการกับลิสต์บ็อกซ์ตั้งแต่การสร้างลิสต์บ็อกซ์ และการจัดการข้อความในลิสต์บ็อกซ์ และอื่นๆ อีกมากมาย แต่ฟังก์ชันที่สำคัญๆ ที่จะมาทำความรู้จักกันมีดังนี้

1) ฟังก์ชันที่ใช้จัดการลิสต์บ็อกซ์ (General Function)

GetCount	คืนค่าจำนวนไอเท็มทั้งหมด
GetTopIndex	คืนตัวเลขอันดับ (index) ของไอเท็มที่แสดงอยู่บนสุดของลิสต์บ็อกซ์
SetTopIndex	กำหนดอันดับให้กับ ไอเท็มที่จะแสดงในตำแหน่งบนสุดของรายการ
GetSel	คืนค่าสถานะของไอเท็มที่เลือกในปัจจุบัน
GetText	คืนค่าของไอเท็ม (ข้อความ) เก็บลงในตัวแปรบัพเฟอร์ (char*)
GetTextLen	คือค่าความยาวของข้อความในลิสต์บ็อกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ฟังก์ชันที่ใช้จัดการเกี่ยวกับการเลือก (Single-Selection Function)

GetCurSel คืนค่าลำดับของไอเท็มที่ถูกเลือก

SetCurSel เลือกไอเท็มที่ต้องการภายในลิสต์บ็อกซ์

3) ฟังก์ชันที่ใช้ในการควบคุมข้อความ (string function)

AddString เพิ่มข้อความไอเท็มลงในลิสต์รายการ

DeleteString ลบข้อความที่ไม่ต้องการออกจากลิสต์บ็อกซ์

InsertString แทรกข้อความที่ต้องการลงในลิสต์บ็อกซ์

ResetContent ลบข้อความไอเท็มทั้งหมดในลิสต์บ็อกซ์

Dir แสดงรายชื่อไฟล์และไดเรกทอรี

FindString ค้นหาข้อความที่ต้องการในลิสต์บ็อกซ์

FindStringExact ค้นหาข้อความแรกที่พบในลิสต์บ็อกซ์

SelecString ค้นหาข้อความและเลือก

2.10.50 การจัดการเมสเสจของลิสต์บ็อกซ์

1) การแมปเมสเสจ LBN_DBLCLK

จะแมปเมสเสจ LBN_DBLCLK เมื่อต้องการให้ลิสต์บ็อกซ์ตอบสนองต่อการ ดับเบิลคลิก ไอเท็มจะทำการแมปโดยการกดปุ่ม <Ctrl+W> เพื่อเรียก MFC CLASSWIZARD ขึ้นมา จากนั้นให้เลือกไปที่ค่า ID ของลิสต์บ็อกซ์ (ซึ่งในตอนนี้ง่าลังแก้ไขโปรเจกต์เวิร์กสเปซ LISTBOX อยู่) ให้เลือกไปที่ปุ่ม IDC_MYLISTBOX จากนั้นในช่อง Message ทางด้านขวาเลือกไปที่ LBN_DBLCLK และกดปุ่ม Add Function เพื่อเพิ่มฟังก์ชัน OnDblclkMylistbox

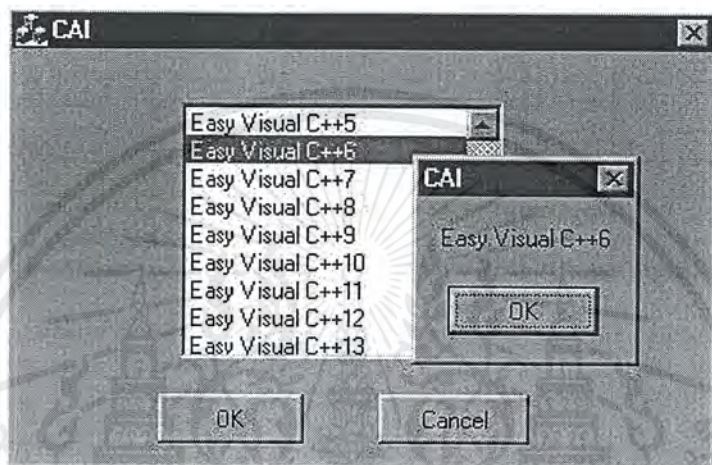
เมื่อเพิ่มฟังก์ชัน OnDblclkMylistbox ลงไปเรียบร้อยแล้ว ให้กดปุ่ม edit code เพื่อทำการแก้ไขโค้ดโปรแกรม ให้เขียนโค้ดดังรูปที่ 2.108 ลงไปในฟังก์ชัน OnDblclkMylistbox

```
void CCaiDlg ::OnDblkMylistbox(){
char buffer[30];
m_cmylistbox.GetText(m_cmylistbox.GetCurSel(),buffer);
MessageBox(buffer);
}
```

รูปที่ 2.108 โค้ด โปรแกรมในการทดสอบการแมปเมสเสจ LBN_DBLCLK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโค้ดโปรแกรมจะเห็นได้ว่า ได้ทำการสร้างบัพเฟอร์ไว้เก็บข้อมูลจำนวน 30 ตัวอักษร และใช้ฟังก์ชัน GetText เพื่อนำเอาข้อความในไอเท็มที่ผู้ใช้เลือกมาเก็บไว้ที่ตัวแปร buffer และแสดงออกทางเมสเสจบ็อกซ์ เพราะฉะนั้นเมื่อ Build โปรแกรมและทดลองรัน ให้ คับเบิลคลิกที่ ไอเท็มในลิสต์บ็อกซ์ ก็จะพบกับเมสเสจบ็อกซ์แสดงข้อความนั้นออกมา ดังรูปที่ 2.109



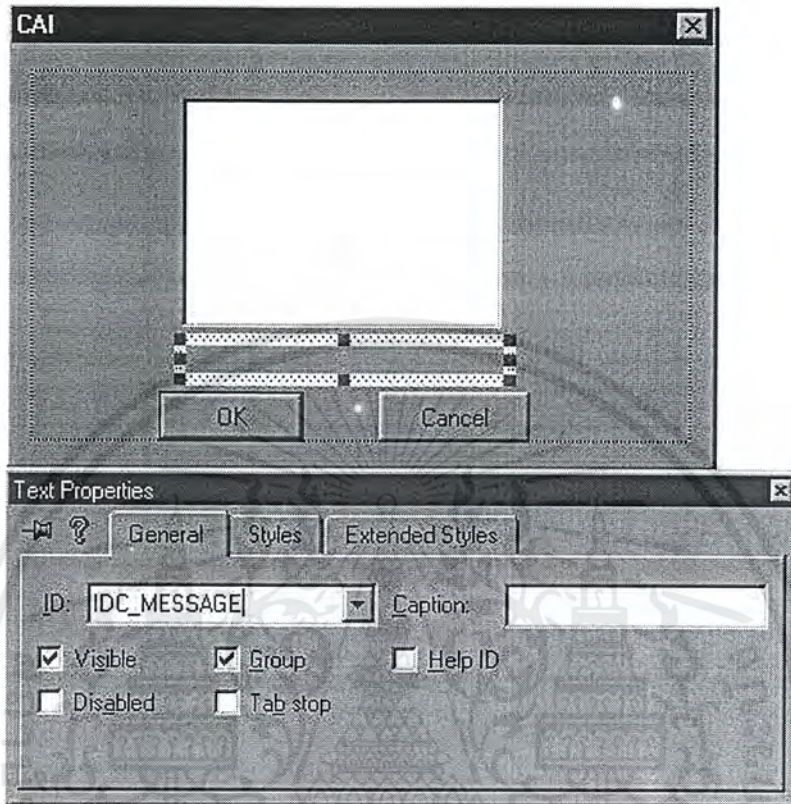
รูปที่ 2.109 ผลการรัน โปรแกรมทดสอบการเมปเมสเสจ LBN_DBLCLK

2) การแมปเมสเสจ LBN_SELCHANGE

เมสเสจ LBN_SELCHANGE เป็นเมสเสจที่จะถูกส่งกลับมาเมื่อมีการเปลี่ยนแปลง การเลือกไอเท็มภายในลิสต์บ็อกซ์ นั่นคือ เมื่อคลิกที่ไอเท็มใดไอเท็มหนึ่งแล้ว และย้ายไปเลือกอีกไอเท็มหนึ่ง เมสเสจนี้ก็จะเกิดขึ้นมาพุดง่าย ๆ ก็คือทุกๆ ครั้งที่มีการเปลี่ยนแปลงการเลือกไอเท็มจะเกิดเมสเสจขึ้น

ให้ทดลองแมปเมสเสจตัวนี้ โดยเรียก ClassWizard ขึ้นมา จากนั้นทำการแมป IDC_MYLISTBOX กับเมสเสจ LBN_SELCHANGE โดย ClassWizard จะสร้างฟังก์ชัน OnSelchangeMylistbox ขึ้นมา

จากนั้นให้ก็เพิ่มสแตติกคอนโทรลลงไป โดยให้ใช้ค่า ID ว่า IDC_MESSAGE โดยกำหนดให้แคปชั่นว่างเปล่าเอาไว้ และวางสแตติกคอนโทรลไว้ด้านล่างของลิสต์บ็อกซ์ ดังรูปที่ 2.110



รูปที่ 2.110 การสร้างสแตติกคอนโทรล IDC_MESSAGE

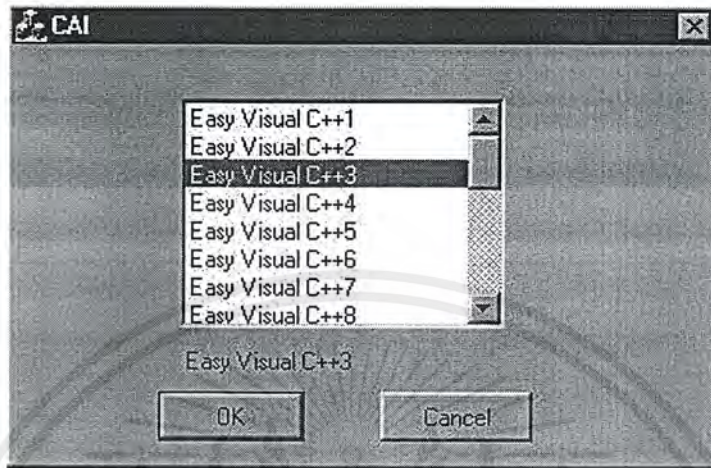
ทำการแก้ไขโปรแกรมดังรูปที่ 2.111

```
void CCaiDlg :: OnSelchangeMylistbox (){
char buffer[30];
m_cmylistbox.GetText(m_cmylistbox.GetCurSel(),buffer);
SetDlgItemText(IDC_MESSAGE , buffer);
}
```

รูปที่ 2.111 โค้ดโปรแกรมการเมปเมสเสจ LBN_SELCHANGE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการรัน โปรแกรมจะได้ผลดังรูปที่ 2.112



รูปที่ 2.112 ผลการรัน โปรแกรมการแม่ปแมสเสจ LBN_SELCHANGE

2.10.51 การจัดการลิสต์บ็อกซ์ด้วย DDX

สามารถนำกระบวนการ DDX มาใช้ในการจัดการข้อมูลของลิสต์บ็อกซ์ได้โดยกระบวนการจัดการข้อมูลต่างๆ นั้นจะทำได้ง่ายขึ้นมาก เพียงเรียกฟังก์ชัน UpdateData เท่านั้น

ก่อนอื่น ให้เพิ่มตัวแปรของลิสต์บ็อกซ์ ที่จะใช้ในกระบวนการ DDX เข้าไปก่อนโดยให้เรียก ClassWizard ขึ้นมา จากนั้นเลือกไปที่แท็บ Member Variables ให้ดับเบิลคลิกที่ IDC_MYLISTBOX ก็จะพบกับไดอะล็อกที่จะให้เพิ่มตัวแปรลงไป ให้ใส่ไปว่า m_data และกำหนดประเภทของตัวแปรเป็น Value และCString ตามลำดับ จากนั้นจึงกดปุ่ม OK

คราวนี้ให้ลองแก้ไขโค้ดโปรแกรมในฟังก์ชัน OnDblclkMylistbox ซึ่งเป็นฟังก์ชัน ที่ใช้รองรับกับแมสเสจ LBN_DBLCLK ที่ได้สร้างเอาไว้ ให้แก้ไขดังรูปที่ 2.113

```
void CCaiDlg ::OnSelchangeMylistbox(){
UpdateData(TRUE);
MessageBox(m_data);
}
```

รูปที่ 2.113 โค้ด โปรแกรมการใช้กระบวนการ DDX ร่วมกับแมสเสจ LBN_DBLCLK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.52 การสร้างและการใช้งานคอมโบบ็อกซ์

คอมโบบ็อกซ์ (Combo Box) เป็นคอนโทรลประเภทเดียวกันกับลิสต์บ็อกซ์ แต่ที่มันเป็นคอมโบบ็อกซ์ก็เพราะว่า มันมีความสามารถเพิ่มขึ้นจากลิสต์บ็อกซ์หลายอย่าง คอมโบบ็อกซ์เป็นคอนโทรลที่มีไว้ให้ผู้ใช้เลือกตัวเลือกที่ต้องการ โดยตัวเลือกที่มีอยู่ในคอมโบบ็อกซ์นั้นจะถูกซ่อนเอาไว้ในคอนโทรลจนกว่าผู้ใช้จะคลิกปุ่มลูกศร

ให้ปฏิบัติตามขั้นตอนต่อไปนี้ เพื่อสร้าง และทดสอบการทำงานของคอมโบบ็อกซ์

1) ขั้นตอนที่ 1 สร้างโปรเจกต์ Combo

ให้สร้างโปรเจกต์ใหม่โดยใช้ AppWizard สร้างโปรแกรมแบบ Dialog Based โดยกำหนดชื่อโปรเจกต์ว่า Combo

2) ขั้นตอนที่ 2 แก้ไขไดอะล็อกรีซอร์ส

เมื่อได้โปรเจกต์เวิร์กสเปซ Combo แก้ไขไดอะล็อกรีซอร์ส IDD_COMBO_DIALOG โดยให้ลบสแตติกคอนโทรล TODO ออกไป และนำคอมโบบ็อกซ์มาวาง

เมื่อนำคอมโบบ็อกซ์มาวางในไดอะล็อกแล้ว จะต้องปรับขนาดของคอมโบบ็อกซ์ด้วย เพราะไม่เช่นนั้นคอมโบบ็อกซ์จะแสดงตัวเลือกไม่ได้ การปรับขนาดทำได้โดยให้คลิกที่ปุ่มลูกศรของคอมโบบ็อกซ์ จากนั้นปรับขนาดของกรอบคอนโทรลให้ยาวลงมา

3) ขั้นตอนที่ 3 กำหนดค่า Id ให้กับคอมโบบ็อกซ์

คราวนี้ ให้กำหนดค่า ID ของคอมโบบ็อกซ์ โดยเรียกไดอะล็อก Combo Box Properties ขึ้นมา จากนั้นให้ใส่ค่า ID เป็น ID_MYCOMBO

4) ขั้นตอนที่ 4 เพิ่มไอเท็มลงในคอมโบบ็อกซ์

สามารถเพิ่มข้อมูล เข้าไปในคอมโบบ็อกซ์ได้ทันที โดยให้เรียกไดอะล็อก Combo Box Properties ขึ้นมา และคลิกไปที่แท็บ Data จากนั้นให้พิมพ์ตัวเลือกที่ต้องการลงไป เมื่อพิมพ์เสร็จบรรทัดหนึ่งแล้วก็กดปุ่ม<Ctrl+ENTER> เพื่อไปยังบรรทัดต่อไป

5) ขั้นตอนที่ 5 ปรับแต่งคอมโบบ็อกซ์

คราวนี้จะมาศึกษาเกี่ยวกับรูปแบบของคอมโบบ็อกซ์กันสักเล็กน้อย ซึ่งการกำหนดรูปแบบของคอมโบบ็อกซ์นั้น สามารถกำหนดตัวเลือกและรูปแบบต่างๆ ของคอมโบบ็อกซ์ได้ในแท็บ Style ของไดอะล็อก Combo Box Properties

ประเภทของคอมโบบ็อกซ์นั้นจะมีอยู่ 3 รูปแบบใหญ่ๆ คือ

- 1) แบบธรรมดา (Simple) เป็นคอมโบบ็อกซ์แบบที่ไม่มีลูกศร จะมีลักษณะกึ่งลิสต์บ็อกซ์ สามารถแก้ไขข้อความที่เลือกได้
- 2) แบบดรอปดาวน์ (dropdown) เป็นคอมโบบ็อกซ์ที่จะซ่อนตัวเลือกเอาไว้ เมื่อต้องการเลือกจะต้องกดปุ่มลูกศรลิสต์บ็อกซ์จึงปรากฏ สามารถแก้ไขข้อความที่เลือกได้
- 3) แบบดรอปลิสต์ (droplist) เป็นคอมโบบ็อกซ์ที่มีลักษณะเดียวกันแบบ Drop down แต่ไม่สามารถแก้ไขข้อความที่เลือกได้ (ใช้ในแท็บคอมโบบ็อกซ์ไม่ได้นั่นเอง)

การเพิ่มข้อความเข้าไปในคอมโบบ็อกซ์ นอกจากจะสามารถกำหนดไอเท็ม โดยใช้ไอคอนเลือก Combo Box Properties ได้แล้ว ยังสามารถใช้ฟังก์ชัน AddString ได้เช่นเดียวกับลิสต์บ็อกซ์ โดยจะต้องสร้างออบเจกต์ของคอมโบบ็อกซ์ขึ้นมาก่อน โดยใช้ ClassWizard

ให้เรียก MFC ClassWizard ขึ้นมาโดยกด <Ctrl+W> จากนั้นก็เลือกแท็บ Member Variable จะพบกับคอนโทรล IDC_MYCOMBO ให้กดปุ่ม Add Variable และกำหนดชื่อของตัวแปรว่า m_cmycombo และกำหนดตัวเลือกในช่อง Category และ Variable Type ให้เป็น Control และ CComboBox

เมื่อได้ออบเจกต์ m_cmycombo แล้ว จะทำการแก้ไขโค้ดโปรแกรมที่ตัวฟังก์ชัน OnInitDialog โดยจะให้ออบเจกต์ m_cmycombo เรียกฟังก์ชัน AddString เพื่อเพิ่มไอเท็ม เข้าไปในลิสต์บ็อกซ์ ดังโค้ด โปรแกรมดังต่อไปนี้ 2.114

```

Bool CComboDlg::OnInitDialog(){
    CDialog::OnInitDialog();
    SetIcon(m_hicon,TRUE);
    SetIcon(m_hicon,FALSE);
    for (int I=1,I<10,I++){
        m_cmycombo.AddString("EASY VISUAL C++");
    }
    return TRUE;
}

```

รูปที่ 2.114 โค้ดโปรแกรมการเพิ่มไอเท็มให้กับคอมโบบ็อกซ์

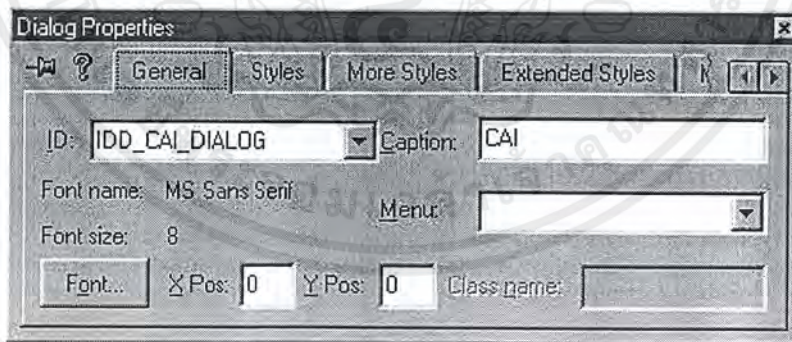
2.10.53 แมสเสจของคอมโบบ็อกซ์

คอมโบบ็อกซ์นั้นมีอยู่ด้วยกัน 3 แบบ คือ แบบธรรมดา(Simple) ,แบบ Drop-down และ Drop-list เพราะฉะนั้นการเลือกใช้แมสเสจ จะต้องดูรูปแบบของคอมโบบ็อกซ์ด้วย

- 1) CBN_DROPDOWN เมื่อผู้ใช้คลิกที่ปุ่มลูกศร เพื่อแสดงรายการไอเท็ม
- 2) CBN_CLOSEUP เมื่อลิสต์รายการไอเท็มถูกเลือกหรือปิดลงไปแล้ว
- 3) CBN_DBLCLK เมื่อมีการดับเบิลคลิกไอเท็มในลิสต์รายการ (Simple)
- 4) CBN_EDITCHANGE เมื่อมีการแก้ไขข้อความในเท็กซ์บ็อกซ์ (Simple และ Drop-Down)
- 5) CBN_EDITUPDATE เมื่อมีการเปลี่ยนแปลงข้อความในเท็กซ์บ็อกซ์ (Simple และ Drop-Down)
- 6) CBN_SELCHANGE เมื่อมีการเลือกที่ไอเท็มตัวใหม่ (เช่นเดียวกับลิสต์บ็อกซ์)

2.10.54 ใดอะล็อก

การปรับแต่งใดอะล็อกด้วยใดอะล็อกอิติเตอร์นั้น ปรับแต่งโดยใช้ตัวเลือกต่างๆ ในใดอะล็อก Properties ให้ลองเปิดใดอะล็อกริชอร์สของโปรเจกต์ที่ได้สร้างกันเอาไว้แล้วขึ้นมาแก้ไข เมื่อเปิดโปรเจกต์ขึ้นมาแล้ว ให้เปิดใดอะล็อกริชอร์สขึ้นมา จากนั้นให้กดปุ่ม <Enter> ก็จะพบกับใดอะล็อก Properties ดังรูปที่ 2.115



รูปที่ 2.115 Dialog Properties

จะเห็นได้ว่า ใดอะล็อก Properties ของใดอะล็อกริชอร์สนั้นมีอยู่ด้วยกันทั้งหมด 5 แท็บ ซ้อนๆ กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) แท็บ General

ตัวเลือกต่างๆ ในแท็บ มีดังนี้

ID	คือค่า ID ของไดอะล็อกที่ใช้ในการอ้างถึงเมื่อแมปแมสเสจ หรือควบคุม
Caption	คือ ข้อความบน ไตเติลบาร์
Menu	คือการกำหนดเมนูริชอร์สให้กับ ไดอะล็อก
Font	เป็นการกำหนดรูปแบบของฟอนต์ให้กับ ไดอะล็อก
Xpos,Ypos	เป็นระยะของจุดเริ่มต้นของการแสดงผล ไดอะล็อก

การเปลี่ยนรูปแบบของฟอนต์ในไดอะล็อกนี้ จะต้องระวังไว้อย่างหนึ่งก็คือ การนำโปรแกรมไปใช้กับเครื่องที่ไม่ได้ติดตั้งฟอนต์แบบที่ใช้ ก็จะไม่สามารถแสดงผลฟอนต์ออกมาได้ แต่อาจจะแสดงเป็นอักขระที่อ่านแล้วไม่รู้เรื่องเป็นต้น การเลือกฟอนต์ที่ใช้ในไดอะล็อกนั้น จะต้องเลือกฟอนต์ที่เป็นมาตรฐาน นั่นคือ ควรจะเป็นฟอนต์ที่มีในทุกๆ ระบบ

2) แท็บ Styles



รูปที่ 2.116 แท็บ Styles

รายละเอียดของตัวเลือกต่างๆที่น่าสนใจ ในแท็บ Styles มีดังนี้

Style	เป็นตัวเลือกรูปแบบของไดอะล็อก ซึ่งจะมีให้เลือกเป็น Overlapped Popup และ Child
Border	เป็นการกำหนดรูปแบบของกรอบไดอะล็อก ซึ่งจะมีให้เลือกตั้งแต่กรอบแบบธรรมดา (Dialog Frame), กรอบแบบปรับขนาดได้ (Resizing), กรอบแบบเล็ก (Thin) และไดอะล็อกไม่มีกรอบ (None)
Title bar	เป็นการกำหนดให้ไดอะล็อกมีไตเติลบาร์

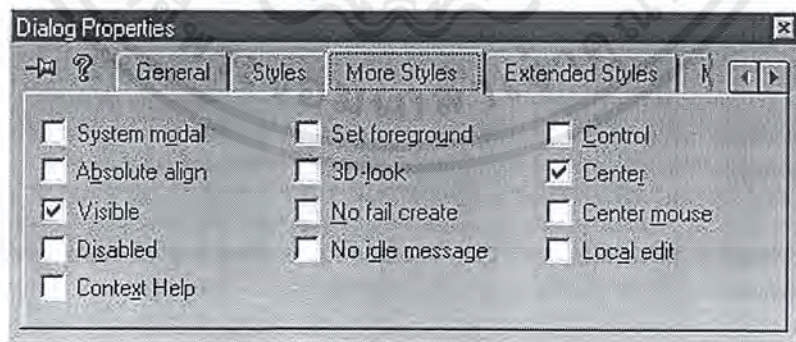
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

System menu	เป็นการกำหนดให้ไอคอนมี System Menu(มี Control Box)
Minimized Box,Maximized Box	เป็นการกำหนดปุ่ม Minimized (ปุ่มยุบเป็นไอคอน) หรือปุ่ม Maximized (ปุ่มขยายเป็นวินโดวส์ขนาดเต็มจอ)ให้กับไอคอน
Horizontal Scroll,Vertical Scroll	เป็นการแสดงสกรอลล์บาร์ในไอคอน ซึ่งจะไม่นิยมใช้สกรอลล์บาร์กัน ยกเว้นในบางกรณีที่ต้องการให้เลื่อน

การกำหนดตัวเลือกเหล่านี้ให้กับไอคอน จะต้องดูความเหมาะสมกันสักเล็กน้อยเพราะถ้าในระบบปฏิบัติการวินโดวส์หนึ่งๆ หรือในไอคอนไม่มีไคเต็ลบาร์ ไอคอนหรือวินโดวส์นั้นก็ไม่สามารถย้ายตำแหน่งได้เลย เพราะการย้ายตำแหน่งของระบบปฏิบัติการวินโดวส์นั้นจะต้องใช้การคลิกเมาส์ที่ไคเต็ลบาร์แล้วลากไปยังตำแหน่งใหม่ที่ต้องการ หรืออีกตัวอย่างหนึ่งก็คือการกำหนดไอคอนให้สามารถ Minimized หรือสามารถยุบเป็นไอคอนได้ก็เป็นข้อดี อีกประการหนึ่งเพราะช่วยในการประหยัดเนื้อที่หน้าจอได้

อาจจะไม่ต้องกำหนดให้ไอคอนมีปุ่ม Maximized ก็ได้ ไม่จำเป็นเลยที่จะให้ระบบปฏิบัติการวินโดวส์แบบไอคอนเป็นวินโดวส์ที่มีขนาดคงที่ ไม่เหมือนกับวินโดวส์แบบเฟรมที่สามารถปรับขนาดได้โดยการไ้เมาส์จับขยาย ถ้าไอคอนมีปุ่ม Maximized เมื่อมีการคลิกที่ปุ่มนี้ ไอคอนก็จะขยายเต็มจอภาพ ผลที่ตามมาก็คือตำแหน่งการวางคอนโทรลในไอคอนจะผิดเพี้ยนไป

3) แท็บ More Styles



รูปที่ 2.117 แท็บ More Styles

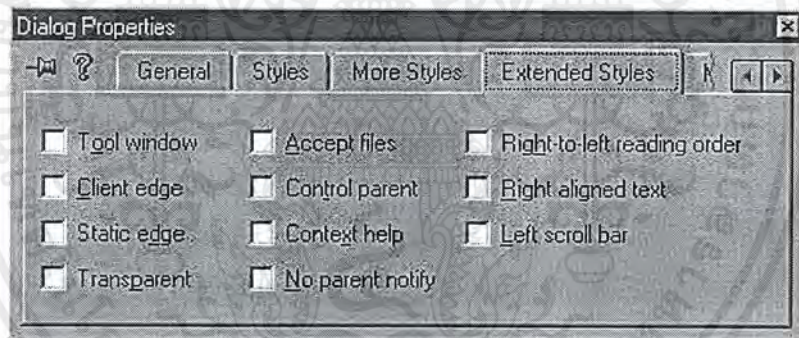
ตัวเลือกต่างๆของแท็บ More Styles ที่น่าสนใจมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

System Modal	สร้างไดอะล็อกแบบ System Modal ซึ่งสามารถสลับงานไปยังงานอื่นได้ในขณะที่ไดอะล็อกทำงาน
Context Help	แสดงเครื่องหมาย ? ไว้ที่มุมขวาของไดอะล็อก สำหรับผู้ใช้สามารถเข้าถึงระบบช่วยเหลือได้ง่าย
3D-Look	แสดงไดอะล็อกให้มีลักษณะ 3D
Center	แสดงไดอะล็อกให้อยู่กึ่งกลางจอภาพ เมื่อเวลารัน
Center mouse	แสดงเมาส์ให้อยู่กึ่งกลางของ ไดอะล็อก เมื่อเวลารัน
Visible	กำหนดให้ไดอะล็อกอยู่ในสถานะปกติ (เป็นค่า Default อยู่แล้ว)

ตัวเลือกที่น่าสนใจก็คือ Center และ Center Mouse ซึ่ง Center เป็นตัวเลือกที่ใช้สำหรับแสดงไดอะล็อกให้อยู่กึ่งกลางจอภาพ และ Center Mouse เป็นตัวเลือกที่ใช้สำหรับแสดงเมาส์ให้อยู่กึ่งกลางของไดอะล็อก (เมื่อรัน โปรแกรมเมาส์จะถูกย้ายตำแหน่งมาที่กึ่งกลางของไดอะล็อกทันที)

4) แท็บ Extended Styles



รูปที่ 2.118 Extended Styles

ตัวเลือกต่างๆในแท็บ Extended Styles ที่น่าสนใจมีดังนี้

Tool Window	ปรับไดอะล็อกให้เป็น Tool Window ก็จะไม่มีปุ่ม Minimized, Maximized จะมีแต่ปุ่ม Close อย่างเดียวที่มุมบนขวา
Client Edge	ปรับให้พื้นที่ไดอะล็อกมีลักษณะยุบลงไป
Static Edge	แสดงกรอบล้อมรอบเล็กๆ
Transparent	ปรับให้พื้นไดอะล็อกเป็นแบบโปร่งใส (ไม่ควรใช้)
Accept Files	ปรับแต่งให้ไดอะล็อกมีความสามารถในการยอมรับไฟล์ที่ลากมาวางจากโปรแกรมอื่นได้ (Drag&Drop)
Context Help	แสดงเครื่องหมาย ? ที่มุมบนขวาของไดอะล็อก

5) แท็บ More Extended Styles

ตัวเลือกต่างๆ ในแท็บ More Extended Styles นี้ ยังไม่มีตัวเลือกใดที่น่าสนใจเมื่อได้รู้จักกับแท็บต่างๆ ของ Dialog Properties กันพอสมควรแล้ว ก็จะมาดูวิธีการใช้งาน และการปรับแต่งไอคอนแบบต่างๆ

2.10.55 การจัดการเมสเสจของไอคอน

เมสเสจที่เกิดขึ้นกับไอคอนมีมากมาย ซึ่งจะมีมากกว่าเมสเสจที่เกิดขึ้นกับคอนโทรล เช่น ปุ่มกด, เท็กซ์บ็อกซ์ หรือลิสต์บ็อกซ์ เป็นต้น เมสเสจที่เกิดขึ้นกับไอคอนนั้นเกิดขึ้นได้ตั้งแต่การเคลื่อนที่ของเมาส์ไปรอบๆ ไอคอน จนถึงการกดคีย์ต่างๆ ในไอคอน จะมาดูกันว่าเมสเสจของไอคอนที่น่าสนใจนั้นมีอะไรกันบ้าง

เมสเสจของไอคอนที่น่าสนใจ และจะศึกษากันมีดังนี้

1) เมสเสจที่เกี่ยวกับเมาส์

WM_RBUTTONDOWN	เมื่อผู้ใช้ดับเบิลคลิกปุ่มขวาในพื้นที่ไอคอน
WM_RBUTTONDOWN	เมื่อมีการกดปุ่มขวาของเมาส์ลง
WM_RBUTTONUP	เมื่อมีการกดปล่อยปุ่มขวาของเมาส์ขึ้น
WM_LBUTTONDOWN	เมื่อผู้ใช้ดับเบิลคลิกปุ่มซ้ายในพื้นที่ไอคอน
WM_LBUTTONDOWN	เมื่อมีการกดปุ่มซ้ายของเมาส์ลง
WM_LBUTTONUP	เมื่อมีการปล่อยปุ่มซ้ายของเมาส์ขึ้น
WM_MOUSEMOVE	เมื่อมีการเคลื่อนที่เมาส์ไปรอบๆ พื้นที่ไอคอน
WM_CONTEXTMENU	เมื่อมีการคลิกเมาส์ปุ่มขวาในพื้นที่ของไอคอน

2) เมสเสจที่เกี่ยวกับคีย์บอร์ด

WM_CHAR	เมื่อมีการกดปุ่มคีย์บอร์ดใดๆ
WM_KEYDOWN	เมื่อมีการกดปุ่มคีย์บอร์ดลง
WM_KEYUP	เมื่อมีการปล่อยปุ่มคีย์บอร์ดขึ้น
WM_HELPINFO	เมื่อผู้ใช้กดปุ่ม F1 เพื่อเรียกระบบช่วยเหลือ

3) เมสเสจที่เกี่ยวกับการสร้างและการสิ้นสุดโปรแกรม

WM_INITDIALOG	เมื่อไอคอนจะถูกแสดงบนจอภาพ
WM_CREATE	เมื่อวินโดวส์ (หรือไอคอน) ถูกสร้าง
WM_PAINT	เมื่อมีการวาดวินโดวส์ใหม่
WM_DESTROY	เมื่อไอคอนถูกทำลายออกจากหน่วยความจำ
WM_CLOSE	เมื่อไอคอนถูกปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) แมสเสจเกี่ยวกับการเปลี่ยนขนาดและตำแหน่ง

WM_MOVE	เมื่อมีการเปลี่ยนตำแหน่งของวินโดวส์ (หรือไดอะล็อก)
WM_SIZE	เมื่อมีการเปลี่ยนแปลงขนาด, Minimized, Maximized และ Restroyed

2.10.56 การติดตั้งเมนูบาร์ และเมนูป๊อปอัพให้กับไดอะล็อก

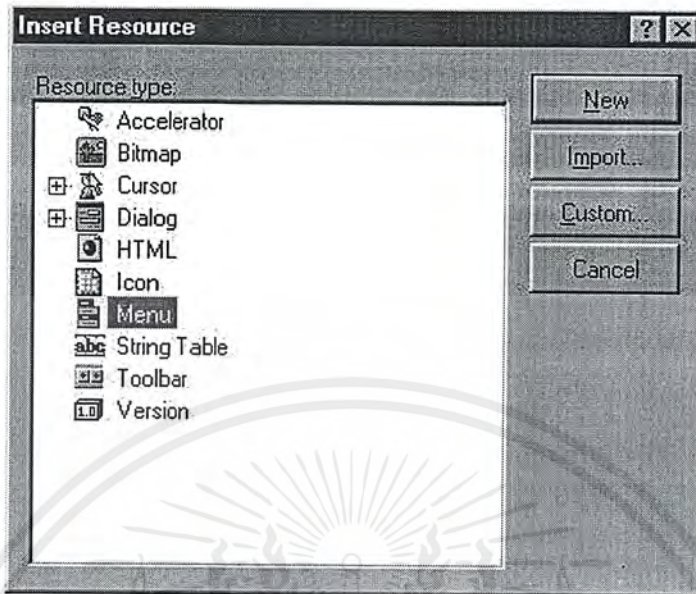
ในการสร้างเมนูให้กับโปรแกรม จะใช้รีซอร์สช่วยในการสร้าง จะเรียกเมนูนี้ว่า เมนูรีซอร์ส (Menu Resource) ซึ่งสามารถใช้เมนูอิดิตเตอร์ (Menu Editer) ซึ่งเป็นอุปกรณ์ตัวหนึ่งที่รวมอยู่ในโปรแกรม Visual C++ 6.0

การใช้งานเมนูอิดิตเตอร์ของโปรแกรม Visual C++ 6.0 นี้จะต่างจากเมนูอิดิตเตอร์ของโปรแกรม Visual Basic หรือโปรแกรมอื่นๆ เพราะรูปแบบการสร้างเมนูของเมนูอิดิตเตอร์นี้จะเป็นการสร้างแบบ WYSIWYG (What you see is what you get) นั่นคือ การสร้างและการแก้ไขจะมีรูปแบบเหมือนกับเมนูจริงๆ

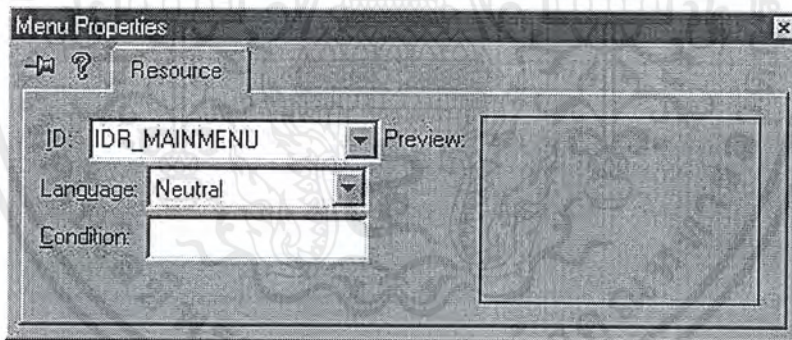
ขั้นตอนต่อไปนี้เป็นสร้างและติดตั้งเมนูให้กับไดอะล็อก (โดยจะใช้โปรเจกต์ Test เป็นโปรเจกต์ทดสอบ)

1) ขั้นตอนที่ 1 สร้างเมนูรีซอร์ส

ขั้นตอนนี้เป็นการสร้างเมนูรีซอร์สใหม่ โดยให้เลือกคำสั่ง Insert>Resource จากนั้นให้เลือกไปที่ Menu และกดปุ่ม New ก็จะพบกับเมนูอิดิตเตอร์ จากนั้นให้เปลี่ยนค่า ID เป็น IDR_MAINMENU ดังรูปที่ 2.119



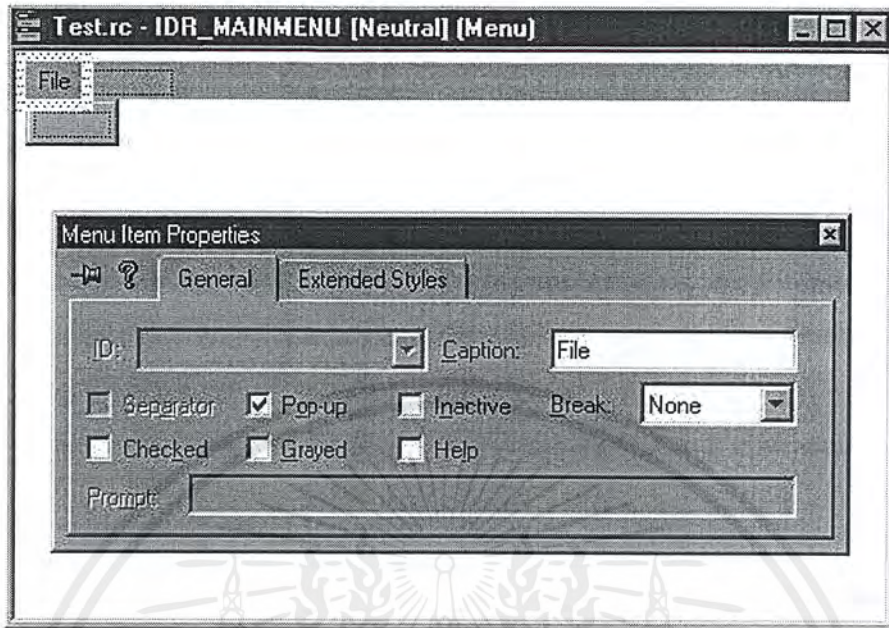
รูปที่ 2.119 การสร้างเมนูริชอร์ส



รูปที่ 2.120 การเปลี่ยนค่า ID ของเมนูริชอร์ส

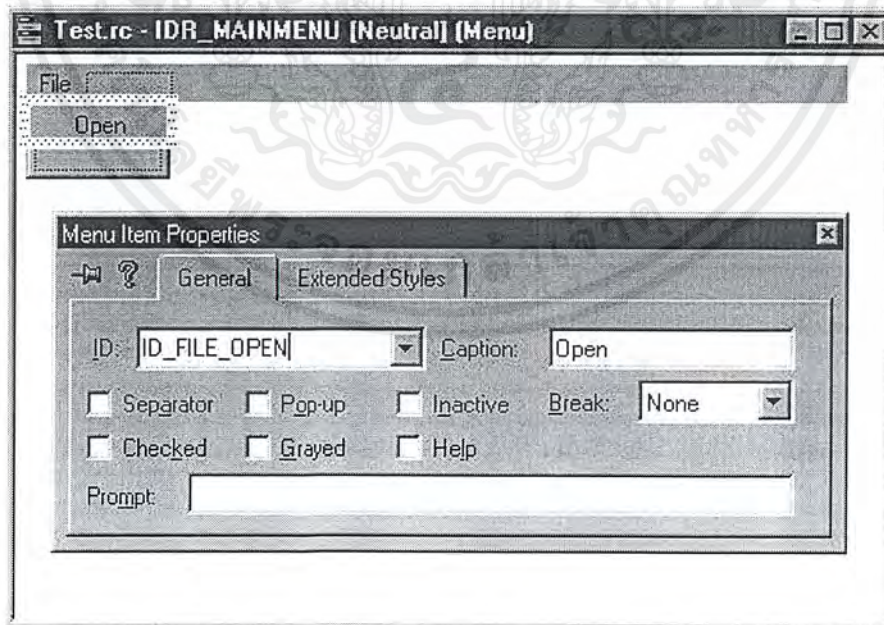
2) ขั้นตอนที่ 2 เพิ่มเมนูป๊อปอัพ

คราวนี้จะทำการเพิ่มคำสั่งต่างๆ ลงไปในเมนู โดยให้คลิกที่ช่องว่างช่องแรกของเมนู และพิมพ์ข้อความว่า File ลงไป ก็จะพบว่าเมนูป๊อปอัพ File จะถูกสร้างขึ้นทันที ดังรูปที่ 2.121



รูปที่ 2.121 การเพิ่มเมนูป๊อปอัพ

จากนั้นจะเพิ่มคำสั่ง Open และ Exit ภายในเมนูป๊อปอัพดังรูปที่ 2.122



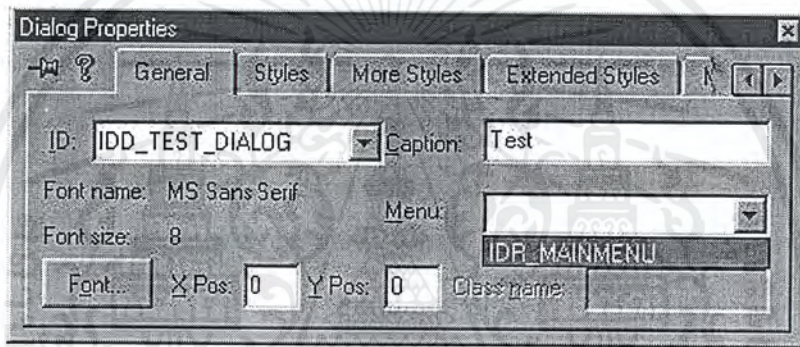
รูปที่ 2.122 การเพิ่มเมนูย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเพิ่มเมนูไอเท็ม Open และ Exit ลงไปแล้ว เมนูอิดิตเตอร์จะกำหนดค่า ID ให้กับเมนู ไอเท็มแต่ละตัวให้โดยอัตโนมัติ โดยเมนู Open จะมีค่า id เป็น ID_FILE_OPEN ส่วนเมนู Exit จะมี ค่า ID เป็น ID_FILE_EXIT ตามลำดับ

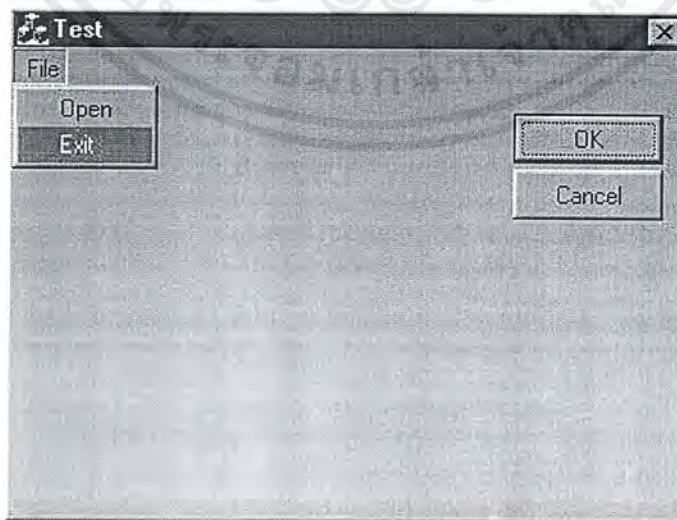
3) ขั้นตอนที่ 3 การติดตั้งเมนู

การติดตั้งเมนู IDR_MAINMENU นี้ให้กับไดอะล็อก โดยให้เปิดไดอะล็อกกริซอร์สที่ต้องการติดตั้งเมนู IDR_MAINMENU นี้ขึ้นมา ในที่นี้จะเปิดไดอะล็อก IDD_TEST_DIALOG จากนั้นเรียกไดอะล็อก DIALOG PROPERTIES และเลือกตัวเลือกในช่อง Menu ไปที่ IDR_MAINMENU ดังรูปที่ 2.123



รูปที่ 2.123 การติดตั้งเมนู

เมื่อได้ทำการติดตั้งเมนูเรียบร้อยแล้ว ให้ทดลอง Build โปรเจกต์เพื่อดูผลการทำงาน



รูปที่ 2.124 ผลการรัน โปรแกรมติดตั้งเมนู

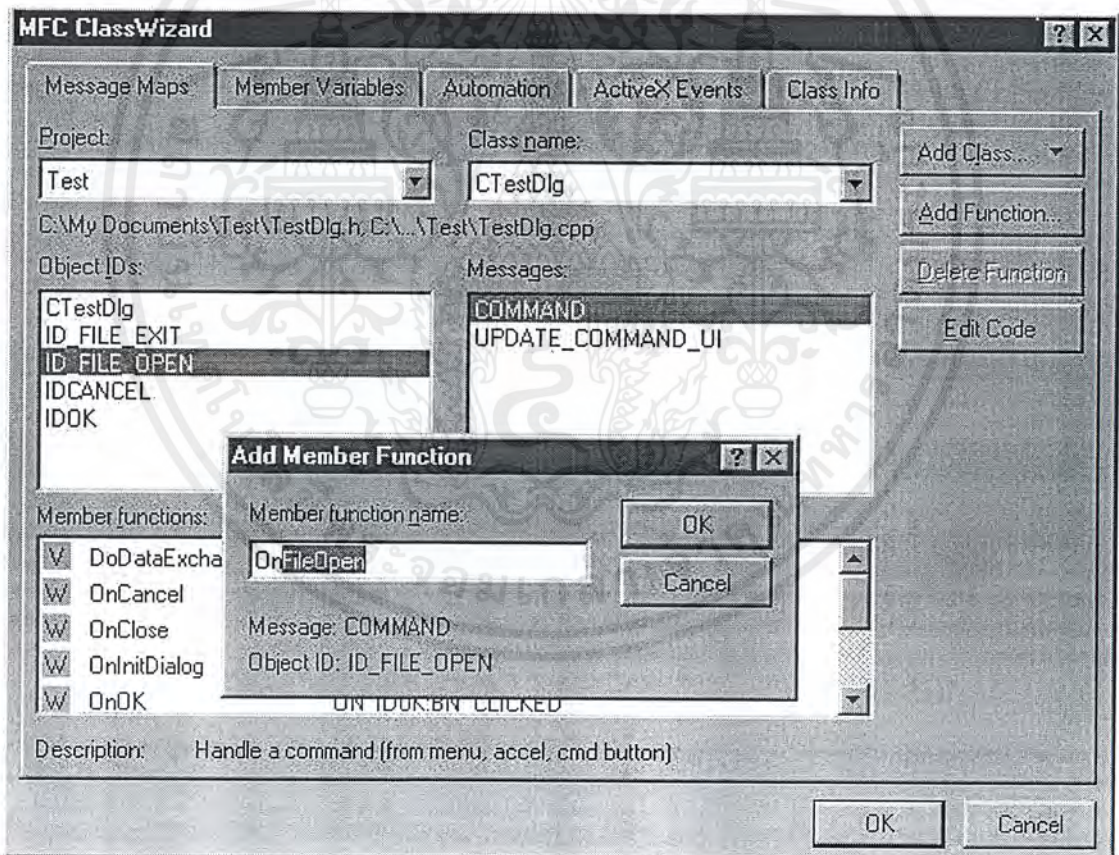
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการรันโปรแกรม จะเห็นได้ว่า เมื่อเลือกเมนู File ก็จะมีปรากฏเมนูไอเท็ม Open และ Exit สามารถเลือกเมนูเหล่านั้นได้ แต่จะยังไม่มีอะไรเกิดขึ้น เพราะเป็นเพียงแค่เมนูเปล่าๆ หากต้องการให้คำสั่ง Open หรือ Exit ใช้การได้ จะต้องทำการแมปแมสเสจให้กับเมนูนั้นเสียก่อน

2.10.57 การใช้ ClassWizard แมปแมสเสจให้กับเมนู

ให้เรียก ClassWizard ขึ้นมา จากนั้นเลือกตัวเลือกในช่อง Class Name ไปที่ CTestDlg จะพบว่าในช่อง Object IDs ปรากฏค่า ID ของเมนูขึ้นมาด้วย ดังรูปที่ 2.125

ให้แมป ID_FILE_OPEN กับ COMMAND จากนั้น ClassWizard จะทำการสร้างฟังก์ชัน OnFileOpen ขึ้นมาให้กับ ให้ดับเบิลคลิกไปที่ฟังก์ชันนี้เพื่อแก้ไขโค้ดโปรแกรม ให้เพิ่มโค้ดโปรแกรมต่อไปนีลงในฟังก์ชัน OnFileOpen



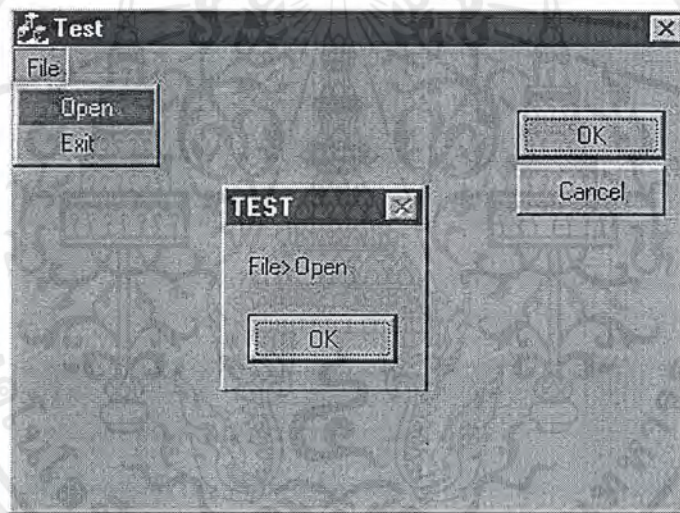
รูปที่ 2.125 การใช้ ClassWizard แมปแมสเสจให้กับเมนู

```
void CtestDlg::OnFileOpen(){
    MessageBox (“File>Open”);
}
```

รูปที่ 2.126 โปรแกรมทดสอบเมนู

เมื่อ Build โปรเจกต์ และรันทดสอบผลการทำงาน เมื่อเลือกเมนู File>Open ก็จะพบกับ
 เมสเสจบ็อกซ์แสดงข้อความว่า File>Open ดังรูปที่ 2.127

สำหรับการแมปเมสเสจ COMMAND ให้กับเมนู ID_FILE_EXIT ก็เช่นเดียวกัน



รูปที่ 2.127 ผลการรัน โปรแกรมทดสอบเมนู

การสร้างเมนูรีซอร์สด้วยเมนูอิดิตเตอร์ สามารถเพิ่มลูกเล่นต่างๆ ลงไปในเมนูได้ เช่น การ
 สร้างเมนูแบบเช็ค หรือการสร้างเส้นกั้นและการดิสเอเบิล (Disable) เมนูต่างๆ สามารถกำหนดรูป
 แบบของเมนูได้โดยใช้ไดอะล็อก Menu Properties

ในการแก้ไขเมื่อนั้น สามารถใช้ปุ่มลูกศรในการเลื่อนตัวเคอร์เซอร์ และใช้ปุ่ม Insert, Delete
 เพื่อแทรก หรือลบเมนูที่ต้องการได้

รายละเอียดของตัวเลือกแต่ละตัวใน Menu Properties ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) เช็คบ็อกซ์ Separator

เช็คบ็อกซ์นี้ ใช้สำหรับการสร้างเส้นกั้นระหว่างเมนูไอเท็มแต่ละตัว โดยสามารถคั่นคำสั่งแต่ละคำสั่งโดยใช้ Separator นี้ โดยให้แทรกเมนูไอเท็มว่างๆ ลงไป โดยการกด Insert ที่เมนู Exit และกาเครื่องหมายที่เช็คบ็อกซ์นี้ จะพบเส้นกั้นระหว่างเมนูไอเท็ม

2) เช็คบ็อกซ์ Popup

เช็คบ็อกซ์ popup นี้ ใช้สำหรับสร้างเมนูป๊อปอัพ ซึ่งจะเป็นเมนูคำสั่งหลัก เช่น พวกเมนู File, Edit, View ซึ่งจะใช้สำหรับแสดงเมนูไอเท็มอื่นๆ ออกมา จึงเรียกว่าเมนูป๊อปอัพ สามารถสร้างเมนู Open ให้เป็นเมนูป๊อปอัพได้ โดยกาเครื่องหมายที่เช็คบ็อกซ์ Popup

3) เช็คบ็อกซ์ Grayed

การเลือกตัวเลือก Grayed เป็นการกำหนดให้เมนูหรือคำสั่งนั้นคิสเอเบิล นั่นคือ เมนูจะเป็นสีเทา และไม่สามารถใช้งานได้

4) เช็คบ็อกซ์ Inactive

ตัวเลือก Inactive จะเป็นการกำหนดให้เมนูหรือคำสั่งนั้นไม่สามารถใช้งานได้ ซึ่งถ้าเลือกตัวเลือก Grayed เอาไว้ ก็เหมือนกับการเลือกตัวเลือก Inactive นั่นเอง แต่ตัวเลือก Inactive จะไม่ทำให้เมนูเป็นสีเทา

5) เช็คบ็อกซ์ Checked

การสร้างเมนูแบบกาเครื่องหมาย สามารถทำได้ โดยเลือกตัวเลือกเช็คบ็อกซ์นี้ จะเห็นได้ว่าเมื่อเลือกตัวเลือกนี้จะมีเครื่องหมาย “ถูก” อยู่หน้าเมนูไอเท็มนั้น

6) เช็คบ็อกซ์ Help

การกาเครื่องหมายที่เช็คบ็อกซ์ Help นี้ เพื่อให้แสดงเมนูนี้ขีดด้านขวาสุดของเมนูบาร์ เช่นเดียวกับการแสดงเมนู Help ของโปรแกรมต่างๆ ไป ซึ่งจะปรากฏอยู่ทางขวาสุด ให้ลองเพิ่มเมนู help ขึ้นมาอีกเมนูหนึ่ง และกาเครื่องหมายที่ช่อง Help

7) เช็คบ็อกซ์ Break

ตัวเลือก Break เป็นการกำหนดรูปแบบของเมนูให้เป็นคอลัมน์ (column) หรือแบบบาร์ (Bar) สามารถกำหนดให้เมนูป๊อปอัพและเมนูไอเท็มปรากฏในแนวนอนหรือแนวตั้งก็ได้ เมนูที่ออกมาจะมีรูปแบบที่แปลกตาออกไป ให้ลองกำหนดตัวเลือก Column ให้กับเมนูไอเท็ม Open กับ Exit

8) เท็คซ์บ็อกซ์ Prompt

เท็คซ์บ็อกซ์ Prompt นี้มีไว้สำหรับพิมพ์ข้อความที่ต้องการให้ปรากฏในสแตตัสบาร์ของระบบปฏิบัติการวินโดวส์แต่ในที่นี้ ไม่ได้สร้างสแตตัสบาร์ เพราะฉะนั้นจึงไม่จำเป็นต้องพิมพ์ข้อความใดๆ ลงในช่อง Prompt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.58 การสร้างเมนูป๊อปอัพโดยการคลิกขวา

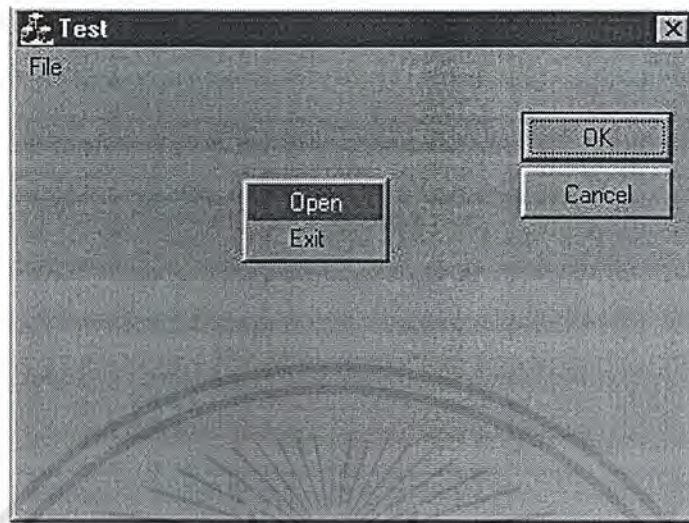
ตอนนี้ได้รู้จักวิธีการสร้างเมนูกันไปแล้ว คราวนี้ จะนำเอาเมนูนี้มาสร้างเป็นเมนูป๊อปอัพกัน โดยจะใช้เมสเสจ WM_CONTEXTMENU ในการแสดงเมนูป๊อปอัพ จะเขียนโค้ดโปรแกรมเพื่อสร้างเมนูป๊อปอัพ โดยใช้เมนูริซอร์ส IDR_MAINMENU เป็นเมนูหลัก โดยก่อนอื่น ให้ทำการแมปเมสเสจ WM_CONTEXTMENU ให้กับไดอะล็อก โดยใช้ ClassWizard เสียก่อน

จากนั้นให้แก้ไขโค้ดโปรแกรมในฟังก์ชัน OnContextMenu ดังรูปที่ 2.128

```
void CTestDlg::OnContextMenu (cwnd*pwnd,cpoint point){
    CMenu main,*popup;
    main.LoadMenu(IDR_MAINMENU);
    popup = main.GetSubMenu(0);
    popup->TrackPopupMenu(TPM_LEFTALIGN|TPM_RIGHTBUTTON,
    point.x,point.y,this);
}
```

รูปที่ 2.128 โปรแกรมทดสอบเมนูป๊อปอัพ

เมื่อเขียนโค้ดโปรแกรมให้กับฟังก์ชัน OnTextMenu ดังในข้างต้นแล้ว ให้ Build โปรเจกต์ และรันโปรแกรมเพื่อดูผลการทำงาน จะพบว่า เมื่อคลิกเมาส์ ปุ่มขวาในบริเวณพื้นที่ไดอะล็อก เมนูป๊อปอัพจะปรากฏ ดังรูปที่ 2.129



รูปที่ 2.129 ผลการรัน โปรแกรมทดสอบเมนูป๊อปอัพ

2.10.59 การสร้างไดอะล็อกใหม่ให้กับโปรแกรม

ในโปรแกรมหนึ่งๆ สามารถสร้างไดอะล็อกให้กับโปรแกรมได้หลายๆ ไดอะล็อก โดยอาจจะสร้างเป็นไดอะล็อก ที่ใช้ในการปรับแต่งโปรแกรม หรือใช้ในการรับข้อมูลจากผู้ใช้ เป็นต้น

การสร้างไดอะล็อกนั้นสามารถทำได้ง่ายๆ โดยใช้ไดอะล็อกอิดิตเตอร์สร้างไดอะล็อกริชอร์สขึ้นมาใหม่ จากนั้นก็จะใช้ ClassWizard จัดการ

1) ขั้นตอนที่ 1 สร้างไดอะล็อกใหม่

เพิ่มไดอะล็อกริชอร์สใหม่ให้กับโปรเจกต์ โดยเรียกเมนู Insert>Resource จากนั้นเลือกไปที่ Dialog และกดปุ่ม New ดังรูป ก็จะพบกับ ไดอะล็อกอิดิตเตอร์ และไดอะล็อกริชอร์สใหม่ที่มีค่า ID เป็น IDD_DIALOG1

2) ขั้นตอนที่ 2 เปลี่ยนค่า ID ของไดอะล็อกใหม่

ให้กดปุ่ม Enter เพื่อเรียกไดอะล็อก Dialog Properties ขึ้นมา จากนั้นให้เปลี่ยนค่า ID ของไดอะล็อกจาก IDD_DIALOG1 ให้เป็น IDD_INPUT_DIALOG

3) ขั้นตอนที่ 3 เพิ่มคลาสใหม่ด้วย ClassWizard

ไดอะล็อกริชอร์ส IDD_INPUT_DIALOG ที่สร้างขึ้นมาใหม่จะยังไม่สามารถใช้งานได้ เพราะยังไม่มีคลาสรองรับ จะต้องสร้างคลาสใหม่ขึ้นมาจัดการกับไดอะล็อกอันใหม่ (ทุกๆ ไดอะล็อกที่สร้างจะต้องมีคลาสเป็นของมันเอง) เมื่อแก้ไขไดอะล็อก IDD_INPUT_DIALOG เสร็จแล้ว ให้อยู่ในหน้าต่างของไดอะล็อกอิดิตเตอร์พร้อมกับเรียก ClassWizard ขึ้นมา โดยกดปุ่ม <Ctrl+W>

เมื่อ ClassWizard ปรากฏจะพบไดอะล็อก Adding a Class

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไดอะล็อก Adding a Class จะถามว่าไดอะล็อก IDD_INPUT_DIALOG ที่สร้างใหม่นี้ ยังมีคลาสที่ใช้จัดการ ต้องการสร้างใหม่หรือไม่ ให้ตอบ OK เพื่อทำการสร้างคลาสใหม่ ก็จะพบกับอีกไดอะล็อกหนึ่งที่มีชื่อ New Class ซึ่งใช้สำหรับสร้างคลาสใหม่

เมื่อกำหนดค่าดังที่ได้อธิบายไว้ครบถ้วนแล้ว ให้คลิกปุ่ม OK คลาส CInputDialog ก็จะถูกสร้างขึ้น และเพิ่มเข้าไปในโปรเจกต์เป็นที่เรียบร้อย ในขั้นตอนนี้ ClassWizard จะสร้างซอร์สโค้ดขึ้นมาด้วยกัน 2 ไฟล์ คือ InputDlg.h และ InputDlg.cpp

ภายในซอร์สไฟล์ทั้งสองนี้ ไฟล์หนึ่งเป็นไฟล์ส่วนหัว (InputDlg.h) ที่ใช้ประกาศคลาสและอีกไฟล์หนึ่งเป็นซอร์สไฟล์ (InputDlg.cpp) ที่ใช้แสดงรายละเอียดของคลาส CInputDialog และเก็บฟังก์ชันต่างๆ

4) ขั้นตอนที่ 4 เพิ่มโค้ดโปรแกรมเพื่อให้เห็นไดอะล็อก

เพิ่มบรรทัด `#include "InputDialog.h"` เข้าไปที่ด้านบนของไฟล์ TestDlg.cpp เพราะไฟล์ InputDlg.h นี้เป็นส่วนหัวของไดอะล็อกที่จะเรียกใช้

จากนั้นให้แก้ไขโค้ดฟังก์ชัน OnFileOpen ที่ได้สร้างเอาไว้แล้ว ดังรูปที่ 2.130

```
void CTestDlg::OnFileOpen(){
    CInputDialog inputdlg;
    inputdlg.DoModal();
}
```

รูปที่ 2.130 โปรแกรมแสดง ไดอะล็อก

เมื่อทำการ Build โปรเจกต์ และรันโปรแกรมเพื่อดูผลการทำงาน ให้เลือกคำสั่ง File-> Open จะพบไดอะล็อก Input Dialog ซึ่งเป็นไดอะล็อกใหม่ที่ได้สร้างขึ้นนั่นเอง

2.10.60 การสร้างและใช้งานสปินคอนโทรล

สปินคอนโทรล หรือ Spin-Button เป็นคอนโทรลที่ใช้สำหรับแสดงค่า และสามารถปรับเปลี่ยนค่าได้ โดยการกดปุ่มลูกศรขึ้นหรือลง โดยมากจะใช้สปินคอนโทรลนี้กับการเลือกค่าที่เป็นตัวเลข คลาสที่ใช้ในการจัดการกับสปินคอนโทรลก็คือคลาส CSpinButtonCtrl

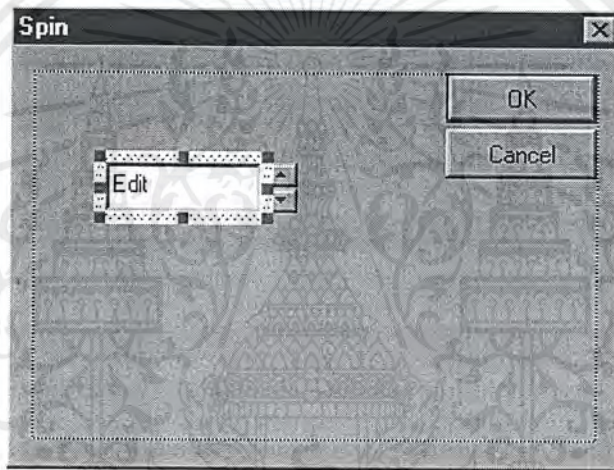
ในการใช้งานสปินคอนโทรลนั้นมีกฎอยู่ว่า สปินคอนโทรลจะต้องจับคู่กับเท็กซ์บ็อกซ์และทำงานร่วมกันเสมอ นั่นคือเมื่อสร้างสปินคอนโทรลขึ้นมา ก็จะต้องสร้างเท็กซ์บ็อกซ์ขึ้นมาด้วยเพื่อใช้สำหรับแสดงค่าที่เปลี่ยนไป

1) ขั้นตอนที่ 1 สร้างโปรเจกต์ Spin

ให้ใช้ AppWizard สร้างโปรเจกต์ใหม่โดยกำหนดชื่อโปรเจกต์ว่า Spin และกำหนดให้โปรเจกต์เป็นโปรเจกต์แบบ Dialog-Based

2) ขั้นตอนที่ 2 นำเท็กซ์บ็อกซ์ และสปีนคอนโทรลมาวาง

ให้เปิดไดอะล็อกบ็อกซ์ของโปรเจกต์ Spin จากนั้นนำเท็กซ์บ็อกซ์และสปีนคอนโทรลจากทูลบาร์มาวางในไดอะล็อก โดยให้สปีนคอนโทรลวางอยู่ทางด้านขวาของเท็กซ์บ็อกซ์ดังรูปที่ 2.131 จากนั้นกำหนดค่า ID ของคอนโทรลโดยให้เท็กซ์บ็อกซ์มีค่า ID เป็น IDC_EDIT และให้สปีนคอนโทรลมีค่า ID เป็น IDC_SPIN



รูปที่ 2.131 การวางเท็กซ์บ็อกซ์และสปีนคอนโทรล

คราวนี้จะใช้คอนโทรลในไดอะล็อก 2 ตัวด้วยกันแล้ว นั่นคือเท็กซ์บ็อกซ์ ซึ่งมีค่า ID เป็น IDC_EDIT และสปีนคอนโทรล ซึ่งมีค่า ID เป็น IDC_SPIN จากนั้นไดอะล็อก Spin Properties ให้คลิกที่แท็บ Styles และกาเครื่องหมายที่เช็คบ็อกซ์ Set buddy integer ดังรูปที่ 2.132 เพื่อกำหนดให้สปีนคอนโทรลแสดงผลเป็นตัวเลข



รูปที่ 2.132 การกำหนดให้เท็กซ์บ็อกซ์ร่วมกับสปินคอนโทรล

3) ขั้นตอนที่ 3 สร้างออบเจกต์ของคอนโทรล

ทำการสร้างออบเจกต์ของคอนโทรลทั้ง 2 ตัว นั่นคือ ออบเจกต์ของเท็กซ์บ็อกซ์ และ สปินคอนโทรล โดยให้กด <Ctrl+W> เพื่อเรียก MFC ClassWizard ขึ้นมาจากนั้นคลิกไปที่แท็บที่ 2 ซึ่งก็คือ Member Variable เพิ่มตัวแปร m_spin ให้กับสปินคอนโทรล โดยกำหนดตัวเลือกในช่อง Category ของตัวแปรทั้ง 2 ให้เป็น Control

4) ขั้นตอนที่ 4 กำหนดบัดดี้ (buddy) ให้กับสปินคอนโทรล

เขียน โค้ด โปรแกรมเพื่อกำหนดบัดดี้ (Buddy) หรือคู่ให้กับสปินคอนโทรล เพื่อใช้ในการแสดงผลข้อมูลหรือตัวเลขที่เปลี่ยนไปดังรูปที่ 2.133

```

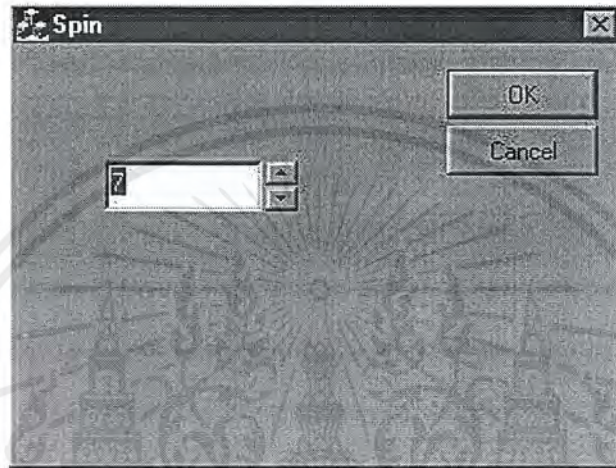
BOOL CSpinDlg::OnInitDialog(){
    CDialog::OnInitDialog();
    SetIcon(m_hIcon, TRUE);
    SetIcon(m_hIcon, FALSE);
    m_spin.SetRange(1,10);
    m_spin.SetBuddy(&m_edit);
    m_spin.SetPos(5);
    return TRUE;
}

```

รูปที่ 2.133 โปรแกรมทดสอบสปินคอนโทรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมไพเลอร์โปรแกรม และทดลองรันเพื่อดูผลทำงานจะพบกับสปินคอนโทรลดังรูปที่ 2.134 เมื่อกดปุ่มลูกศรของสปินคอนโทรลเพื่อเปลี่ยนค่าจะเห็นว่าตัวเลขที่อยู่ในเท็กซ์บ็อกซ์เปลี่ยนไปด้วย แต่จะได้สูงสุดแค่ 10 และต่ำสุดแค่ 1 เท่านั้น เพราะกำหนดค่าต่ำสุด และสูงสุดไว้ที่ 1 และ 10 นั้นเอง



รูปที่ 2.134 ผลการรัน โปรแกรมสปินคอนโทรล

2.10.61 การนำค่าในสปินคอนโทรลไปใช้

การนำค่าที่เปลี่ยนแปลงในสปินคอนโทรลไปใช้นั้น สามารถใช้กระบวนการ DDX จัดการได้ โดยจะต้องสร้างตัวแปร DDX ของเท็กซ์บ็อกซ์ขึ้นมาหนึ่งตัวเพื่อใช้เก็บค่าที่เปลี่ยนแปลงไป

ให้เรียก MFC ClassWizard ขึ้นมา จากนั้น จากนั้นคลิกไปที่แท็บ Member Variable และดับเบิลคลิกที่ IDC_EDIT เพื่อเพิ่มตัวแปร m_value อีกตัวหนึ่ง โดยกำหนดให้เป็นตัวแปรแบบจำนวนเต็ม (int)

จากนั้นให้เลือกไปที่แท็บ Message Maps ให้แมปเมสเสจ BN_CLICKED ให้กับปุ่ม IDOK จะได้ฟังก์ชัน OnOk

ให้เพิ่มโค้ดโปรแกรมดังรูปที่ 2.135 ลงในฟังก์ชัน OnOk โดยโค้ดโปรแกรมที่จะเพิ่มลงไป นี้ เป็นการแสดงให้เห็นการนำเอาค่าที่เปลี่ยนแปลงไปในสปินคอนโทรลไปใช้

```

Void CSpinDlg::OnOk()
{

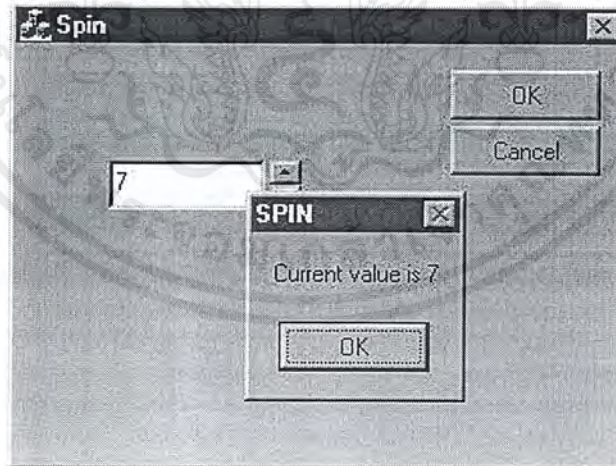
UpdateData(true);
CString msg;
Msg.Format("Current value is %d",m_value);
MessageBox(msg);

}

```

รูปที่ 2.135 โปรแกรมตรวจสอบค่าในสปินคอนโทรล

เมื่อแก้ไขโค้ดโปรแกรมในฟังก์ชัน OnOk เรียบร้อยแล้ว ให้คอมไพล์โปรแกรมและทดลองรันเพื่อทดสอบผลการทำงาน ให้เปลี่ยนค่าในสปินคอนโทรลไปเรื่อยๆ และกดปุ่ม OK จะพบกับไดอะล็อกแมสเสจบ็อกซ์แสดงค่าในปัจจุบันออกมา ดังรูปที่ 2.136



รูปที่ 2.136 ผลการทำงานการตรวจสอบค่าในสปินคอนโทรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.62 การสร้างและการใช้งานสไลเดอร์คอนโทรล

1) ขั้นตอนนี้ที่ 1 สร้างโปรเจกต์ Slider

ให้สร้างโปรเจกต์ใหม่โดยใช้ AppWizard โดยกำหนดให้เป็นโปรเจกต์แบบ Dialog-Based เหมือนเดิม และกำหนดชื่อโปรเจกต์ให้เป็น Slider

2) ขั้นตอนนี้ที่ 2 สร้างสไลเดอร์คอนโทรล

ให้เปิดไดอะล็อกบ็อกซ์ IDD_SLIDER_DIALOG และนำสไลเดอร์คอนโทรลมาวางลงในไดอะล็อกและกำหนดค่า ID ให้กับสไลเดอร์คอนโทรลเป็น IDC_SLIDER และนำแท็กชื่อบ็อกซ์วางลงในไดอะล็อกโดยกำหนดค่า ID ของแท็กชื่อบ็อกซ์เป็น IDC_EDIT

3) ขั้นตอนนี้ที่ 3 สร้างออบเจกต์ของคอนโทรล

ให้เรียก MFC ClassWizard ขึ้นมาเพื่อเพิ่มออบเจกต์และตัวแปรของคอนโทรล ดังนี้ สร้างออบเจกต์ m_slider ให้กับสไลเดอร์คอนโทรล โดยกำหนด Category เป็น Control สร้างตัวแปร m_slidervalue ให้กับสไลเดอร์คอนโทรล โดยกำหนดให้เป็นตัวแปรแบบจำนวนเต็ม

สร้างตัวแปร m_value ให้กับแท็กชื่อบ็อกซ์ IDC_EDIT โดยกำหนดให้เป็นตัวแปรแบบจำนวนเต็ม

4) ขั้นตอนนี้ที่ 4 กำหนดค่าเริ่มต้นให้กับสไลเดอร์คอนโทรล

ให้เปิดไฟล์ SliderDlg.cpp จากนั้นเลือกไปที่ฟังก์ชัน OnInitDialog เพื่อเพิ่มโค้ดโปรแกรมดังรูปที่ 2.137 ลงไป

```

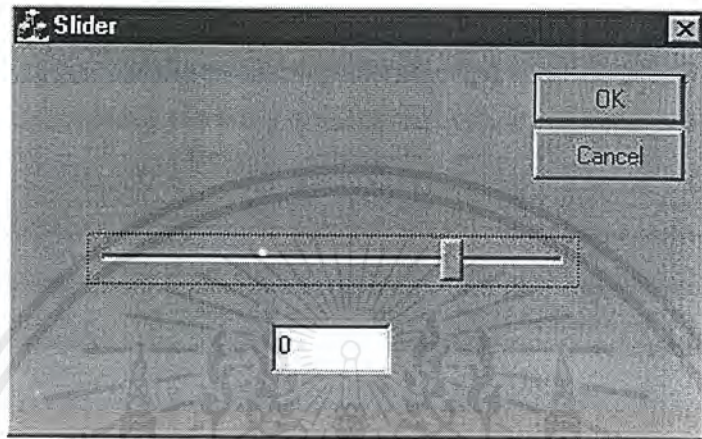
BOOL CSliderDlg::OnInitDialog(){
    CDialog::OnInitDialog();
    SetIcon(m_hicon,TRUE);
    SetIcon(m_hicon,FALSE);
    m_slider.SetRange(1,10);
    return TRUE;
}

```

รูปที่ 2.137 โปรแกรมทดสอบสไลเดอร์คอนโทรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทดลองคอมไพล์และรันโปรแกรม จะพบกับไดอะล็อกดังรูปที่ 2.138 ให้ทดลองเลื่อนตัวเลื่อนของสไลเดอร์คอนโทรลดู จะสังเกตได้ว่าตัวเลื่อนจะเลื่อนไปที่ละ 1 จากซ้ายไปขวา หรือจากขวาไปซ้ายทั้งหมด 10 ค่าตั้งแต่ 1 จนถึง 10



รูปที่ 2.138 ผลการรัน โปรแกรมทดสอบสไลเดอร์คอนโทรล

ขั้นตอนที่ 5 เพิ่มการอัปเดตให้กับเท็กซ์บ็อกซ์

เขียนโค้ดโปรแกรมเพื่ออัปเดตค่าให้กับเท็กซ์บ็อกซ์ สไลเดอร์คอนโทรลจะต่างจากสปินคอนโทรลตรงที่สปินคอนโทรลจะจับคู่กับเท็กซ์บ็อกซ์อัตโนมัติ แต่สไลเดอร์คอนโทรลจะไม่มีการจับคู่บังคับ เพราะฉะนั้น จึงต้องเขียนโค้ดโปรแกรมเพื่อทำการอัปเดตอัตโนมัติ

การอัปเดตสไลเดอร์คอนโทรลนั้น จะใช้เมสเสจที่ชื่อ WM_HSCROLL เพราะว่าเมื่อมีการเลื่อนตัวเลื่อนของสไลเดอร์คอนโทรลเมสเสจ WM_HSCROLL จะถูกส่งกลับมา เพราะฉะนั้น จะต้องแมปเมสเสจที่ชื่อ WM_HSCROLL เสียก่อน โดยให้เรียก MFC ClassWizard และดับเบิลคลิกไปที่ WM_HSCROLL ฟังก์ชัน OnHScroll เพิ่มลงในคลาส

จากนั้นให้แก้ไขโค้ดโปรแกรมของฟังก์ชัน OnHScroll ดังรูปที่ 2.139

```

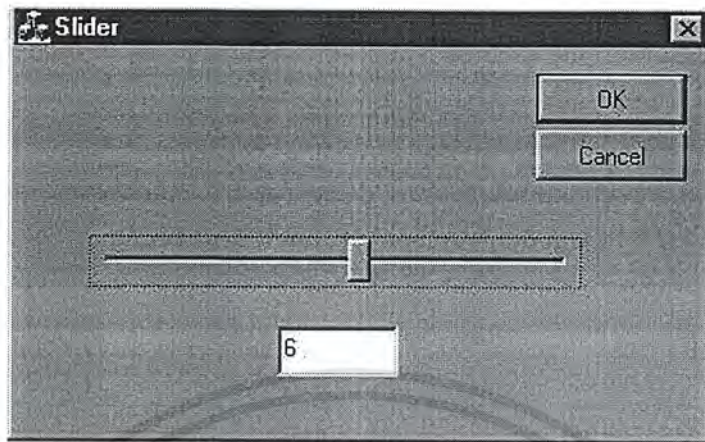
void CSliderDlg ::OnHScroll(UINT nSBCode,UINT nPos,CScrollBar*PScrollBar){
    CSliderCtrl*s;
    s=(CSliderCtrl*)pScrollBar;
    if (s==&m_slider){
        UpdateData(TRUE);
        m_value=m_slidervalue;
        UpdateData(FALSE);
    }
    CDialog::OnHScroll(nSBCode,nPos,pScrollBar);
}

```

รูปที่ 2.139 โปรแกรมทดสอบสไลเดอร์คอนโทรลร่วมกับเท็กซ์บ็อกซ์

จากโค้ดโปรแกรมในข้างต้น บรรทัดแรก ได้ประกาศตัวแปร s ให้เป็นออบเจกต์ของคลาส CSliderCtrl จากนั้นได้ตรวจสอบออบเจกต์ s ว่าใช้สไลเดอร์คอนโทรล m_slider หรือไม่ (โดยใช้เงื่อนไข if) ถ้าใช่ก็ทำการอัปเดตค่าในสไลเดอร์คอนโทรล โดยฟังก์ชัน UpdateData(TRUE) โดยค่าที่อัปเดตนี้ได้เก็บไว้ในตัวแปร m_slidervalue และทำการถ่ายค่ามายังตัวแปร m_value ซึ่งเป็นตัวแปร แบบจำนวนจริงของเท็กซ์บ็อกซ์ IDC_EDIT จากนั้นทำการ UpdateData (FALSE) ค่าในเท็กซ์บ็อกซ์จึงเปลี่ยนแปลงตามการเลื่อนสไลเดอร์คอนโทรลนั่นเอง

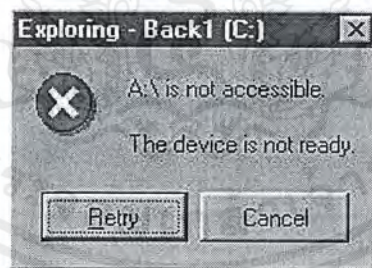
เมื่อรันโปรแกรม และทดลองเลื่อนตัวเลื่อนจะพบว่าค่าภายในเท็กซ์บ็อกซ์ได้มีการเปลี่ยนแปลงไปตามการเลื่อนของตัวเลื่อน ดังรูปที่ 2.140



รูปที่ 2.140 ผลการรันโปรแกรมทดสอบสไลเดอร์คอนโทรลร่วมกับเท็กซ์บ็อกซ์

2.10.63 แมสเสจบ็อกซ์

แมสเสจบ็อกซ์เป็นไดอะล็อกมาตรฐานในระบบวินโดวส์ที่ใช้แสดงข้อความเตือน หรือแสดงข้อความช่วยเหลือ และยังสามารถตอบโต้กับผู้ใช้ได้อีกด้วย สามารถพบเห็นแมสเสจไดอะล็อกเหล่านี้ ได้โดยวิธีง่ายๆ เพียงแค่เรียกโปรแกรม Window Explorer ขึ้นมา จากนั้นก็คลิกไปที่ไดรฟ์ A: แต่ไม่ต้องไปใส่แผ่นในไดรฟ์ รอสักครู่ก็จะมีข้อความปรากฏขึ้นดังรูปที่ 2.141



รูปที่ 2.141 ตัวอย่างการเกิดแมสเสจบ็อกซ์

แมสเสจบ็อกซ์ประเภทนี้เพื่อใช้แสดงข้อความข่าวสาร หรือตอบโต้กับผู้ใช้ในโปรแกรม หรือ แอปพลิเคชันของ ได้โดยวิธีง่ายๆ ซึ่ง MFC ได้เตรียมฟังก์ชันที่ใช้สำหรับสร้างไดอะล็อกแมสเสจบ็อกซ์ไว้แล้ว สามารถเรียกใช้ได้ทันที เมื่อต้องการฟังก์ชันนี้ก็คือ MessageBox

รูปแบบของฟังก์ชัน MessageBox มีรูปแบบ และพารามิเตอร์ที่คล้ายๆ กัน ดังนี้

int MessageBox (“ข้อความที่แสดง”, “ข้อความบนไตเติลบาร์” และ “รูปภาพ และปุ่มกด”)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของพารามิเตอร์มีดังนี้





- 1) “ข้อความที่ต้องการแสดง” ก็คือข้อความที่ต้องการแสดงภายในเมสเสจบ็อกซ์ ต้องเป็นข้อมูลแบบสตริง เช่น Cstring หรือ char*
- 2) “ข้อความในไต่เต็ลบาร์” คือข้อความที่ปรากฏบนไต่เต็ลบาร์ของเมสเสจบ็อกซ์ หากไม่ต้องการกำหนดข้อความบนไต่เต็ลบาร์ สามารถกำหนดพารามิเตอร์ตัวนี้ให้เป็น NULL ได้
- 3) รูปภาพและปุ่มกด เป็นการกำหนดรูปแบบของปุ่มกด และรูปภาพที่แสดงในเมสเสจบ็อกซ์ (รูปไอคอนเล็กๆ) โดยจะต้องใส่มาร์โครที่ MFC ได้กำหนดไว้ และกันด้วยเครื่องหมาย(pipe) ระหว่างมาร์โครแต่ละตัว ซึ่งมาร์โครจะมีอยู่ด้วยกัน 2 แบบ คือ มาร์โครของปุ่มกด และมาร์โครของรูปไอคอน ซึ่งมีดังนี้

1) มาร์โครของปุ่มกด

MB_ABORTRETRYIGNORE	แสดงปุ่ม Abort ,retry และ Ignore
MB_OK	แสดงปุ่ม OK ปุ่มเดียว
MB_OKCANCEL	แสดงปุ่ม OK และ CANCEL
MB_RETRYCANCEL	แสดงปุ่ม Retry และ Cancel
MB_YESNO	แสดงปุ่ม Yes และ No
MB_YESNOCANCEL	แสดงปุ่ม Yes ,No และ Cancel

2) มาร์โครของรูปไอคอน

MB_ICONASTERISK,MB_ICONINFORMATION	คำแนะนำ
MB_ICONHAND,MB_ICONSTOP	หยุด
MB_ICONQUESTION	คำถาม
MB_ICONEXCLAMATION	ข้อความเตือน

	MB_ICONHAND, MB_ICONSTOP, and MB_ICONERROR
	MB_ICONQUESTION
	MB_ICONEXCLAMATION and MB_ICONWARNING
	MB_ICONASTERISK and MB_ICONINFORMATION

รูปที่ 2.142 ลักษณะของมาร์โครรูปไอคอนต่าง ๆ

การเรียกใช้เมสเสจบ็อกซ์ สามารถเขียนไว้ในส่วนใดของโปรแกรมก็ได้ ซึ่งก็แล้วแต่เหตุการณ์ที่ต้องการ เช่นต้องการให้โปรแกรมถามความแน่ใจจากผู้ใช้ก่อนจบโปรแกรมก็จะเรียกฟังก์ชัน MessageBox เพื่อถามความแน่นอนจากผู้ใช้เสียก่อน ซึ่งจะเห็นได้ว่ามีหลากหลายวิธี

ฟังก์ชัน MessageBox เป็นฟังก์ชันที่จะมีการคืนค่ากลับ (รีเทิร์น) ซึ่งที่คืนกลับมาจะเป็นค่าของปุ่มกดที่ผู้ใช้เลือก

ค่าคงที่ที่รีเทิร์นกลับมาจากฟังก์ชัน MessageBox มีดังต่อไปนี้

IDABORT	ปุ่ม Abort ถูกกด
IDCANCEL	ปุ่ม Cancel ถูกกด
IDIGNORE	ปุ่ม Ignore ถูกกด
IDNO	ปุ่ม No ถูกกด
IDOK	ปุ่ม OK ถูกกด
IDRETRY	ปุ่ม Retry ถูกกด
IDYES	ปุ่ม Yes ถูกกด

2.10.64 การจัดการจัดการเท็กซ์ไฟล์โดยใช้คลาสจาก MFC

คลาสที่ใช้สำหรับจัดการไฟล์ใน MFC ที่ใช้มากที่สุด มีอยู่ 2 คลาส คือ

- 1) คลาส CFile เป็นคลาสแม่ของคลาสที่ใช้ในการจัดการไฟล์ทั้งหมดของ MFC
- 2) คลาส CStdioFile สืบทอดจาก CFile ใช้สำหรับจัดการเท็กซ์ไฟล์โดยเฉพาะ

การใช้คลาส CStdioFile บอกได้ว่าเป็นการเรียกใช้ฟังก์ชันจากคลาสที่ตรงไปตรงมา สามารถเขียนในรูปของคลาส CString และอ่านข้อมูลมาเก็บอยู่ในรูปของคลาส CString ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดและเมมเบอร์ฟังก์ชันของคลาส CStdioFile มีดังนี้

m_pStream	เป็นตัวแปรพอยเตอร์ที่จะชี้ตำแหน่งปัจจุบันของไฟล์
CstdioFile()	เป็นฟังก์ชันคอนสตรัคเตอร์ของคลาส CStdioFile
ReadString()	อ่านข้อมูลจากไฟล์
WriteString()	เขียนข้อมูลลงไฟล์

รูปแบบการประกาศออบเจกต์ของคลาส CstdioFile มีดังรูปที่ 2.143

CStdioFile (“ชื่อไฟล์”,รูปแบบของการเปิดไฟล์);

รูปที่ 2.143 รูปแบบการประกาศออบเจกต์ของคลาส CStdioFile

คอนสตรัคเตอร์ของคลาส CStdioFile นั้นต้องการพารามิเตอร์ 2 ตัวด้วยกัน ตัวแรกก็คือ พาทของไฟล์ที่ต้องการเปิด และตัวที่สองก็คือ รูปแบบของการเปิดไฟล์ ซึ่งคลาส CStdioFile นี้เป็น คลาสที่สืบทอดมาจากคลาส CFile เพราะฉะนั้น การกำหนดรูปแบบของการเปิดไฟล์จึงต้องอาศัย ค่าแฟลกที่กำหนดเอาไว้ในคลาส CFile ดังนี้

CFile::modeCreate	บอกถึงการเปิดไฟล์ใหม่ (สร้างไฟล์ใหม่เลย)
CFile::modeRead	บอกถึงการเปิดไฟล์ที่มีอยู่เพื่ออ่าน
CFile::modeWrite	บอกถึงการเปิดไฟล์ที่มีอยู่เพื่อเขียนข้อมูล
CFile::modeReadWrite	บอกถึงการเปิดไฟล์เพื่ออ่านและเขียนข้อมูล

2.10.65 การอ่านข้อมูลจากเท็กซ์ไฟล์

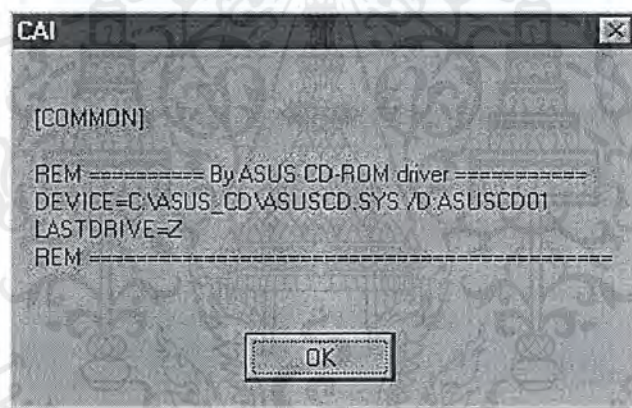
สมมติว่าต้องการเปิดไฟล์ CONFIG.SYS ขึ้นมาเพื่ออ่านข้อมูล จะต้องเขียนโค้ดโปรแกรม ดังรูปที่ 2.144

```

CstdioFile s("C:\\config.sys",CFile::modeRead);
CString buff,msg;
while( !feof(s.m_pStream)){
s.ReadString(buff);
msg+=buff;
msg+="\n";
}
MessageBox (msg);

```

รูปที่ 2.144 ตัวอย่าง โปรแกรมการอ่านเท็กซ์ไฟล์



รูปที่ 2.145 ผลการรัน โปรแกรมการอ่านเท็กซ์ไฟล์

จากโปรแกรมในรูปที่ 2.144 เป็นการสร้างออบเจกต์ของ CStdioFile ก็คือ s ให้เปิดไฟล์ CONFIG.SYS ซึ่งอยู่ในไดเรกทอรี C: โดยเปิดอ่านแบบ modeRead หรือแบบอ่านอย่างเดียว จากนั้นเข้าสู่ลูป while และอ่านข้อมูลจากไฟล์จนสิ้นสุดไฟล์ โดยตรวจสอบตัวแปร m_pStream ซึ่งเป็นตัวชี้ตำแหน่งของไฟล์ว่าเป็น NULL หรือไม่ โดยใช้ฟังก์ชัน feof() ตรวจสอบ

ฟังก์ชัน ReadString จะอ่านข้อมูลในแต่ละบรรทัดของไฟล์มาเก็บไว้ในตัวแปร buff จากนั้นให้เก็บลงตัวแปร msg และปิดท้ายด้วย \n เป็นการสั่งให้ขึ้นบรรทัดใหม่ เมื่ออ่านข้อมูลครบก็จะหลุดออกจากลูป while และแสดงผลข้อมูลที่อยู่ในตัวแปร msg ซึ่งก็คือไฟล์ CONFIG.SYS นั่นเอง

2.10.66 การบันทึกข้อมูลลงในไฟล์

การบันทึกข้อมูลลงไฟล์สามารถทำได้โดยใช้ฟังก์ชัน `WriteString()` ซึ่งการเขียนข้อมูลนั้นสามารถใช้ข้อมูลที่อยู่ในรูปออบเจกต์ `CString` หรือ `char*` ก็ได้ เพราะโอเวอร์โหลดฟังก์ชัน `WriteString()` ยอมให้ใช้พารามิเตอร์คนละประเภทได้ ตัวอย่างต่อไปนี้แสดงการข้อมูลลงไฟล์ `DATA.TXT`

```
CStdioFile file ("data.txt",Cfile::modeCreate|CFile::modeWrite);
file.WriteString("Hello this is a test\n");
file.WriteString ("Visual C++ is easy for you");
```

รูปที่ 2.146 โปรแกรมทดสอบการบันทึกไฟล์

เมื่อรันโค้ดโปรแกรมในข้างต้น โปรแกรมจะทำการสร้างไฟล์ `DATA.TXT` ขึ้นมาใหม่ (`ModeCreate`) และกำหนดให้เป็น โหมดเขียนไฟล์ (`modeWrite`) จากนั้นทำการเขียนประโยคลงไป 2 ประโยค

บทที่ 3

การออกแบบและการทำงาน

โปรแกรมนี้มีเป้าหมายที่จะนำเสนอโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0 โดยการใช้โปรแกรม Visual C++ 6.0 ซึ่งภายในของโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0 นี้ ประกอบไปด้วยเรื่อง การติดตั้งโปรแกรม, ความรู้พื้นฐานเกี่ยวกับภาษา C++, อาร์เรย์, พอยเตอร์, การใช้แบบฟอร์มและเครื่องมือในโปรแกรม Visual C++ 6.0 และการติดต่อกับเท็กซ์ไฟล์

โปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0 ได้จัดขึ้น เพื่อให้ผู้เรียน ได้เรียนรู้เนื้อหาด้วยตนเองตามความสามารถ ความสนใจ และความพร้อมของผู้เรียนเอง ซึ่งเป็นจุดประสงค์หลักที่ต้องการจะนำไปใช้แทนการสอนแบบเดิม

3.1 เครื่องมือที่ใช้ในการออกแบบสร้างโปรแกรมช่วยสอน

3.1.1 ทางด้านฮาร์ดแวร์

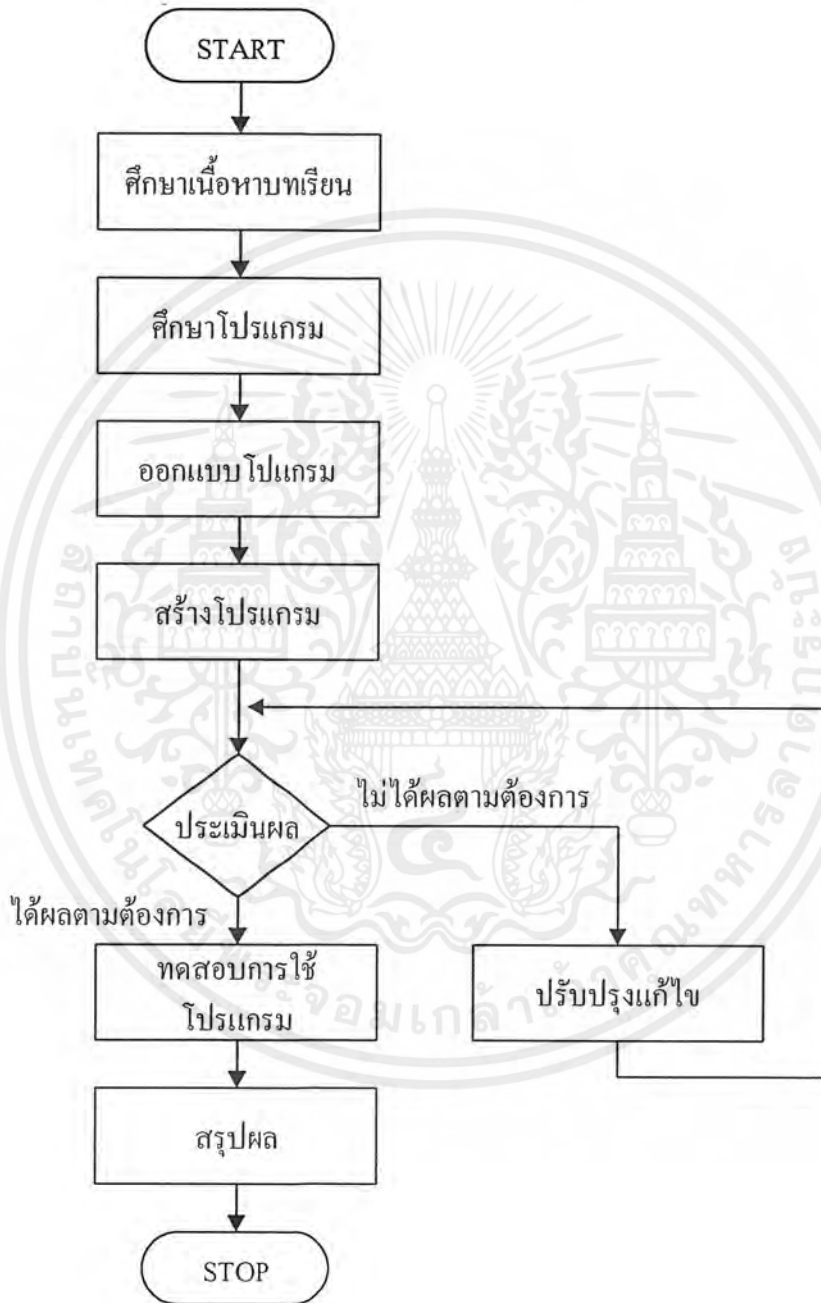
- 1) เครื่องคอมพิวเตอร์ Pentium II 350 MHz
- 2) หน่วยความจำ (RAM) 16 Mbytes
- 3) ฮาร์ดดิสก์ (HARD DISK) 21.4 Gbytes
- 4) เมาส์ (MOUSE)
- 5) ไมโครโฟน (Microphone)
- 6) CD-ROM Drive

3.1.2 ทางด้านซอฟต์แวร์

- 1) โปรแกรม Microsoft Visual C++ 6.0
- 2) โปรแกรม Microsoft PowerPoint 97
- 3) โปรแกรม Lotus Screen Cam 97

3.2 ขั้นตอนการออกแบบโปรแกรม

ขั้นตอนการออกแบบโปรแกรม สามารถแสดงได้ดังรูปที่ 3.1

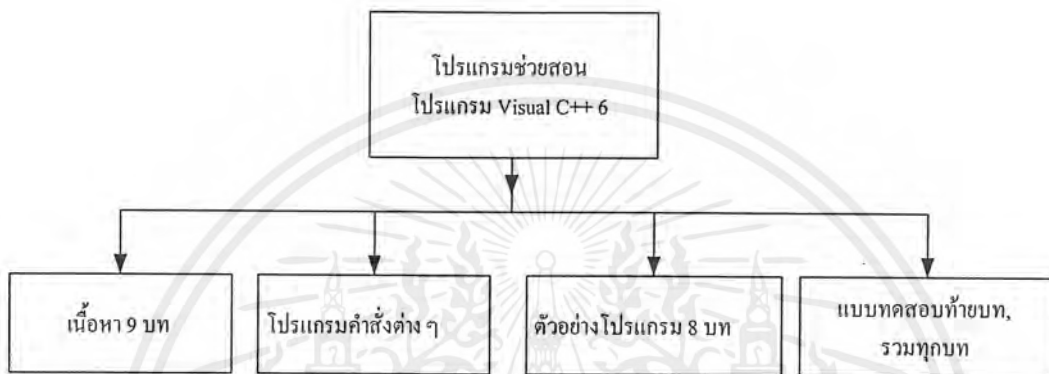


รูปที่ 3.1 ผังงานขั้นตอนการออกแบบโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 โครงสร้างของโปรแกรม

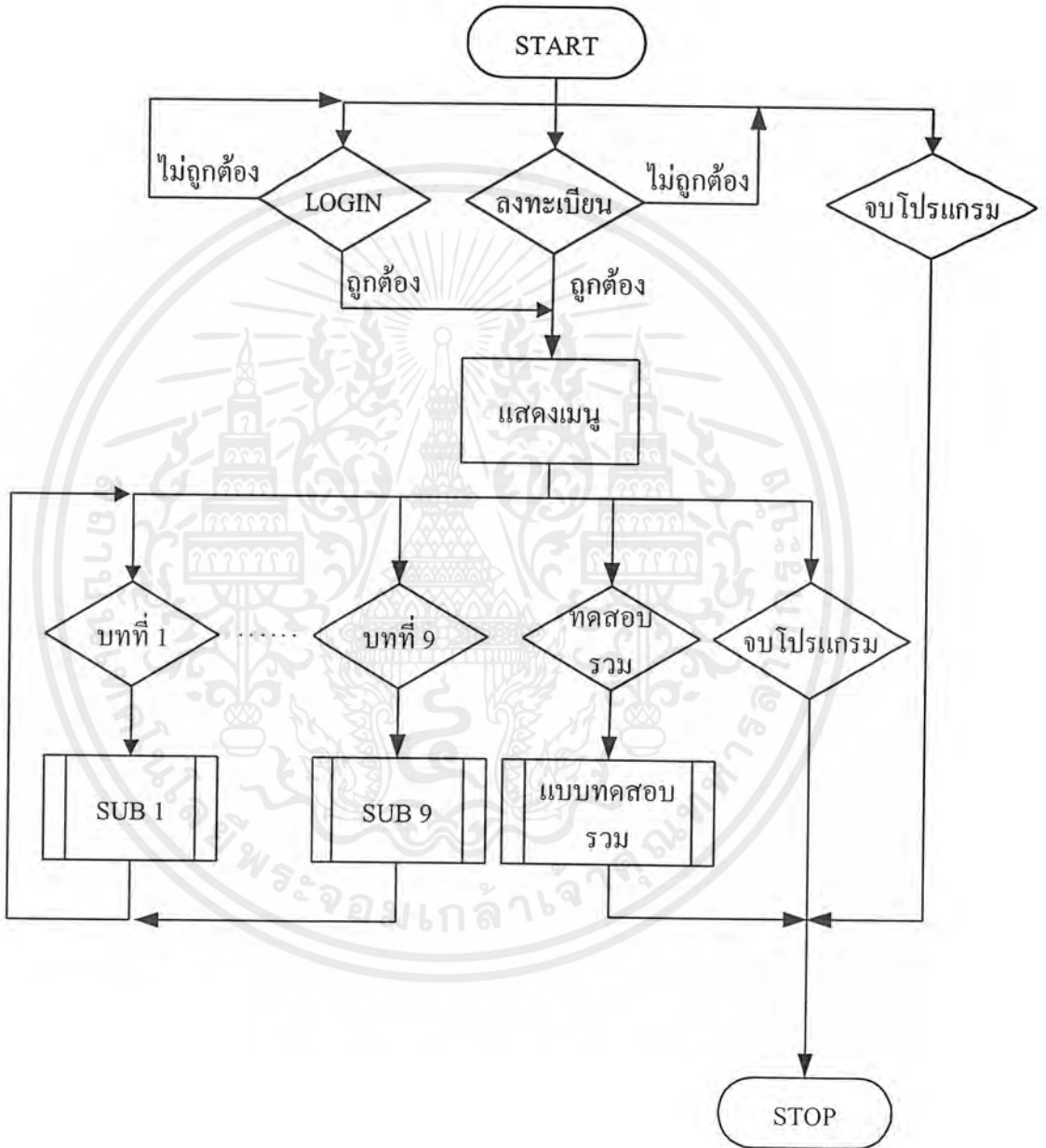
โครงสร้างของโปรแกรมช่วยสอนนี้ได้แบ่งออกเป็นส่วน ๆ ได้ 4 ส่วน เพื่อให้การทำงานเกิดความเป็นระเบียบ สามารถแบ่งได้ดังนี้ คือ เนื้อหา, โปรแกรม, ตัวอย่างโปรแกรม และแบบฝึกหัด สามารถแสดงได้ดังรูปที่ 3.2



รูปที่ 3.2 แผนผังโครงสร้างของโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0

3.4 โครงสร้างของโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0

โครงสร้างโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0 สามารถแสดงได้ดังรูปที่ 3.3

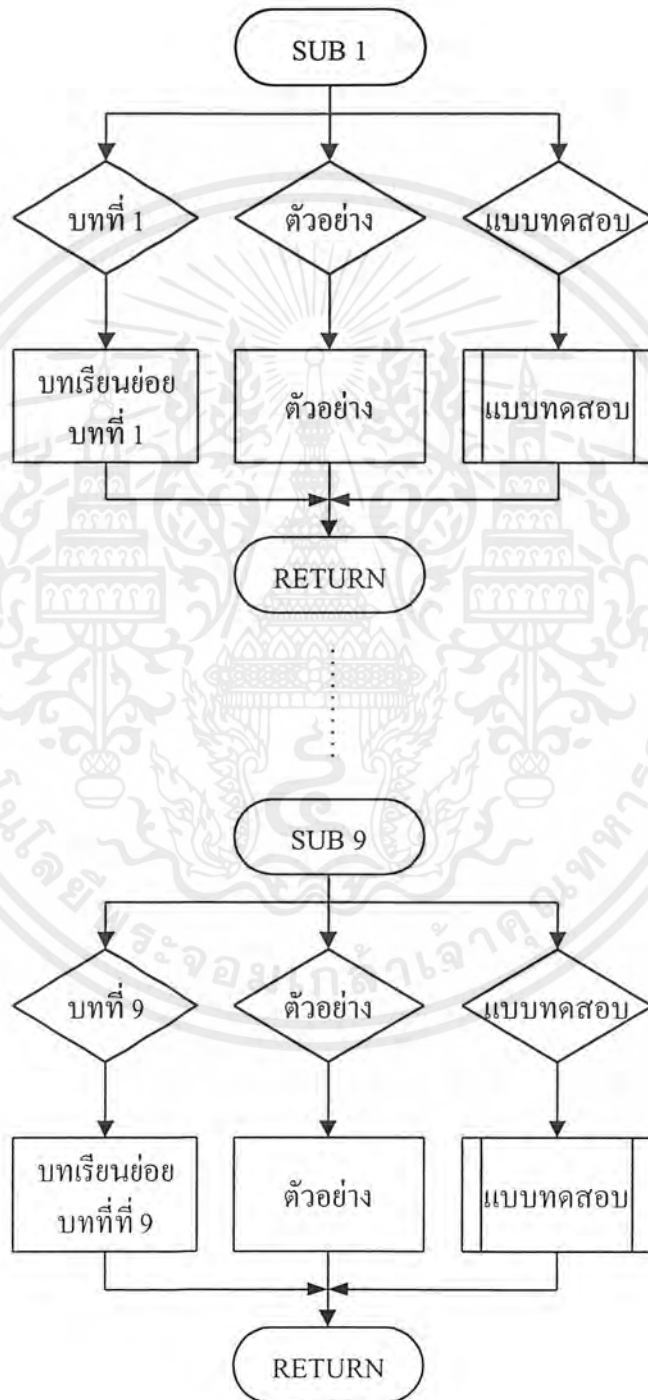


รูปที่ 3.3 ฟังงาน โปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 บทเรียน

ผังงานของบทเรียนแสดงดังรูปที่ 3.4

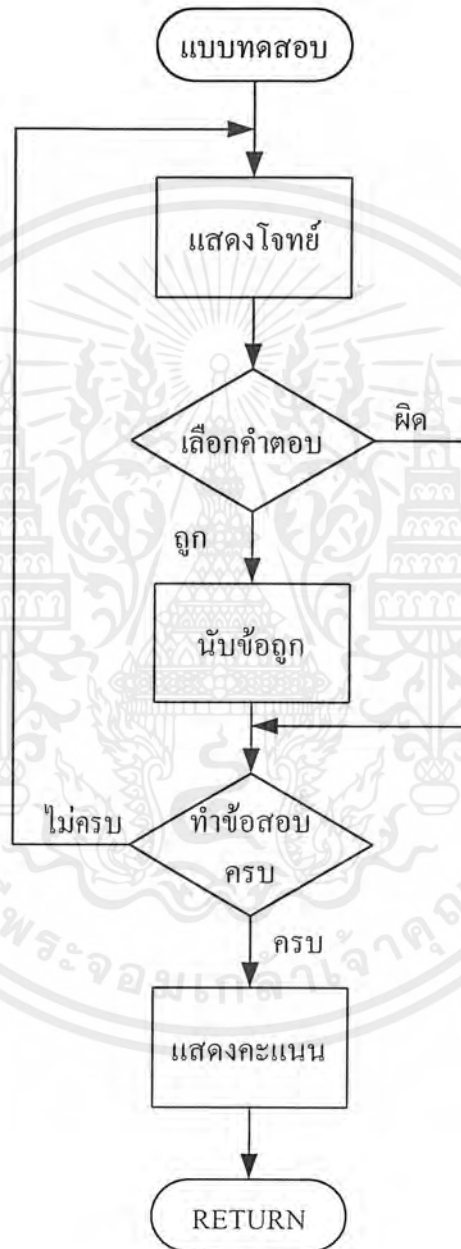


รูปที่ 3.4 ผังงานของบทเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 แบบทดสอบ

ผังงานแบบทดสอบแสดงได้ดังรูปที่ 3.5

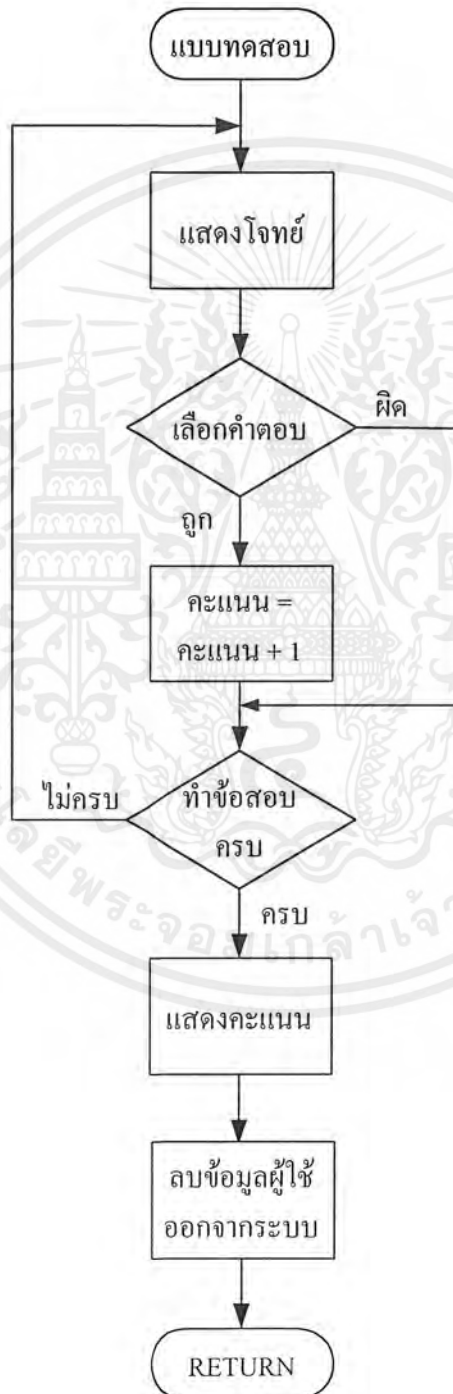


รูปที่ 3.5 ผังงานแบบทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 แบบทดสอบรวม

ผังงานแบบทดสอบรวมแสดงได้ดังรูปที่ 3.6



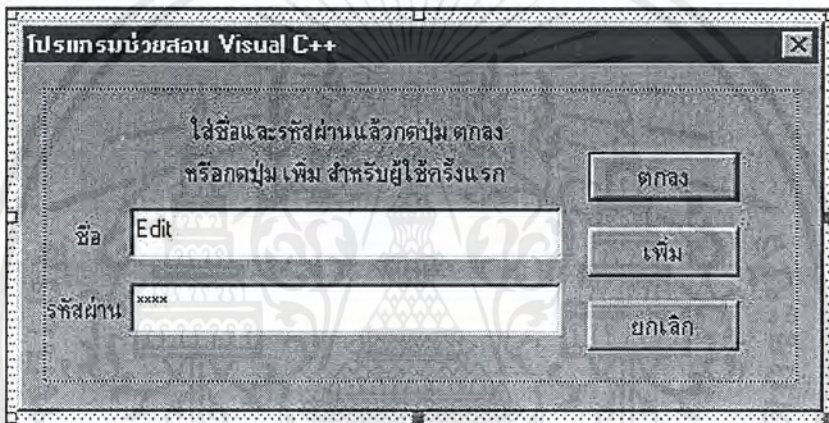
รูปที่ 3.6 ผังงานแบบทดสอบรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 การสร้างโปรแกรม

การสร้างโปรแกรมช่วยสอนนี้ ใช้โปรแกรม Visual C++ 6.0 เขียนโปรแกรมหลักในการจัดการเกี่ยวกับบทเรียนและแบบทดสอบทั้งหมด และใช้โปรแกรม ScreenCam เรียกใช้ไฟล์ที่เป็นภาพเคลื่อนไหวพร้อมคำบรรยาย ที่มีนามสกุลเป็น .SCM ในการให้โปรแกรม ScreenCam ทำงานนั้น จะใช้โปรแกรม Visual C++ 6.0 เขียนโปรแกรมเพื่อเรียกโปรแกรม ScreenCam ขึ้นมารัน

สำหรับการทำงานของโปรแกรมนั้น เมื่อโปรแกรม CAI เริ่มทำงานจะต้องมีการ Login ก่อน จึงได้ทำการออกแบบไดอะล็อกของการ Login ดังรูปที่ 3.7



รูปที่ 3.7 ไดอะล็อกของการ Login

กำหนดให้ Text Box ในส่วนของรหัสผ่านมีรูปแบบแสดงผลแบบ Password ซึ่งทำได้โดยการเลือก Edit Properties เป็น Password ดังรูปที่ 3.8

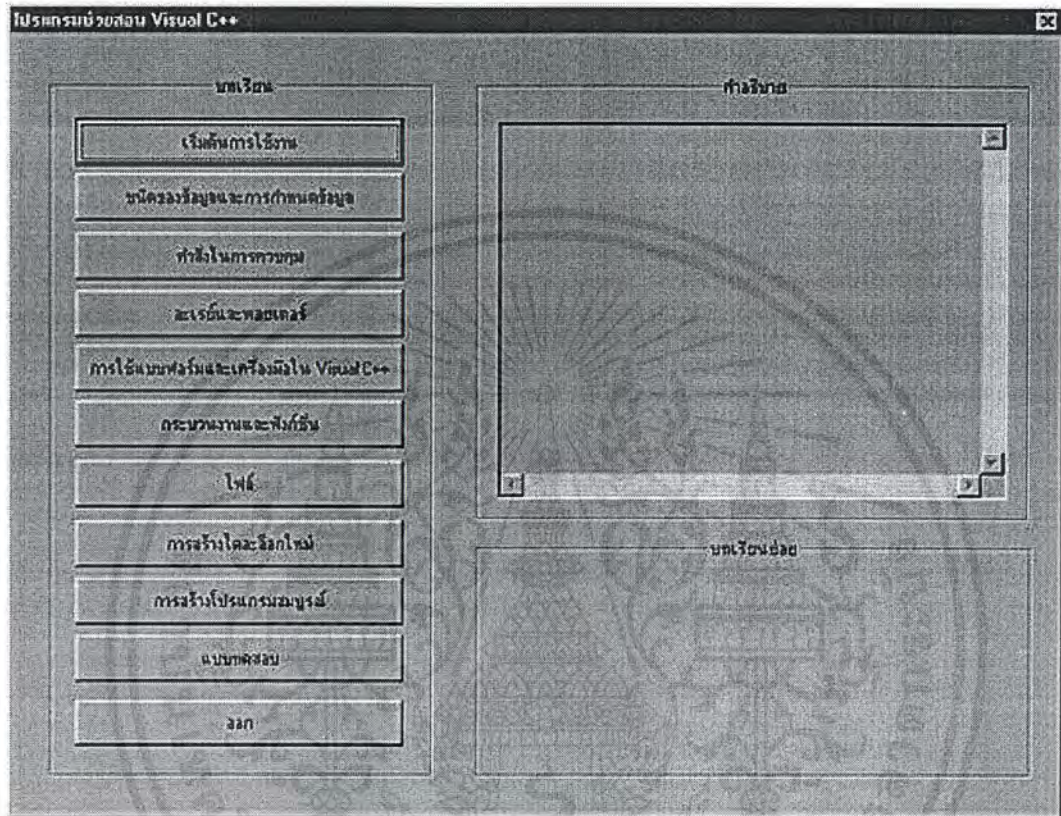


รูปที่ 3.8 การกำหนดรูปแบบให้ Text Box เป็นแบบ Password

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.1 การสร้างเมนูหลัก

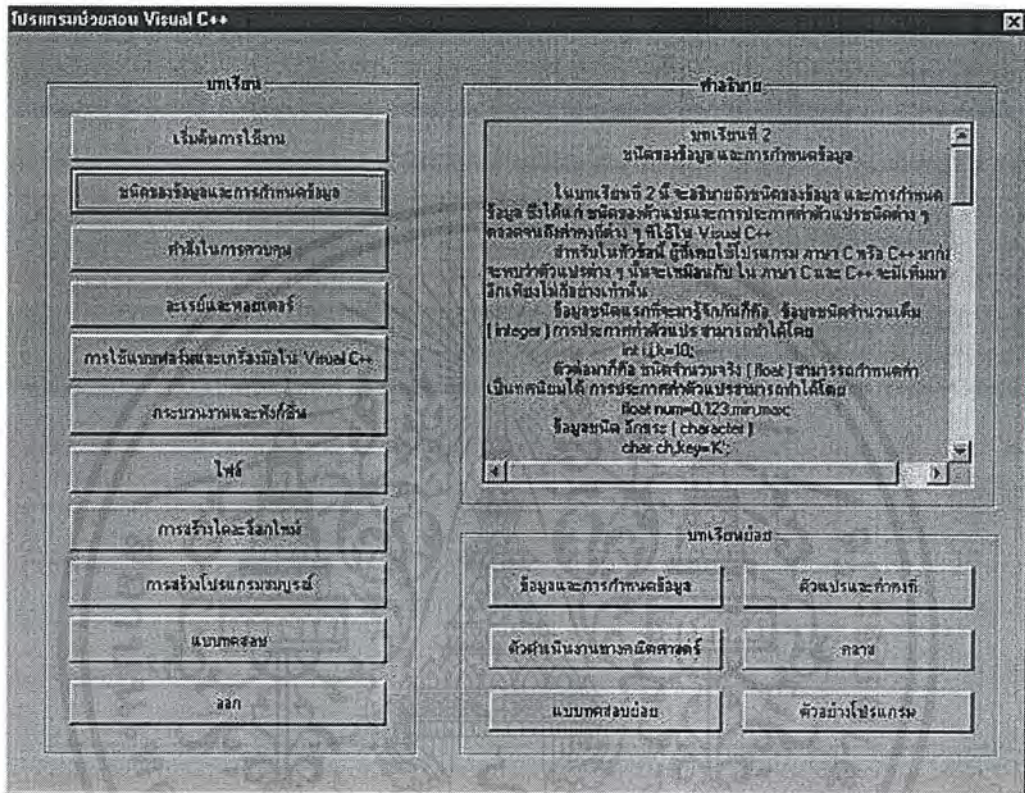
เมื่อทำการ Login ผ่านแล้วจึงจะเข้าสู่ Menu หลัก ได้ออกแบบไดอะล็อกดังรูปที่ 3.9



รูปที่ 3.9 เมนูหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีการเลือกบทเรียนแล้วจะต้องเข้าสู่บทเรียนย่อย ดังนั้นจึงมีการออกแบบให้แสดงเมนูย่อยในการเข้าสู่บทเรียนย่อยและแสดงรายละเอียดของบทเรียน ดังรูปที่ 3.10



รูปที่ 3.10 แสดงรายละเอียดของบทเรียน และเมนูย่อย

เพื่อให้สามารถแสดงเมนูย่อยได้จึงต้องทำการเขียน โค้ด โปรแกรมให้แสดงปุ่มกดเพิ่มเมื่อมีการเลือกบทเรียน ดังรูปที่ 3.13 และโค้ด โปรแกรมในการแสดงคำอธิบายแสดงได้ดังรูปที่ 3.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Cbutton *but1,*but2,*but3,*but4,*but5,*but6;
Cfont *font;

    delete but1;

    delete but2;

    delete but3;

    delete but4;

    delete but5;

    delete but6;

    but1=new Cbutton();
    but2=new Cbutton();
    but3=new Cbutton();
    but4=new Cbutton();
    but5=new Cbutton();
    but6=new Cbutton();

    but1->create
    (“ข้อมูลและการกำหนดข้อมูล”,WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON,crect
    (350,380,515,410),this,IDC_TT1);

    but2-> create
    (“ตัวแปรและค่าคงที่”,WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON,crect
    (530,380,695,410),this,IDC_TT2);

    but3->create
    (“ตัวดำเนินงานทางคณิตศาสตร์”,WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON,crect
    (350,425,515,455),this,IDC_TT3);

    but4->create

    but6->create

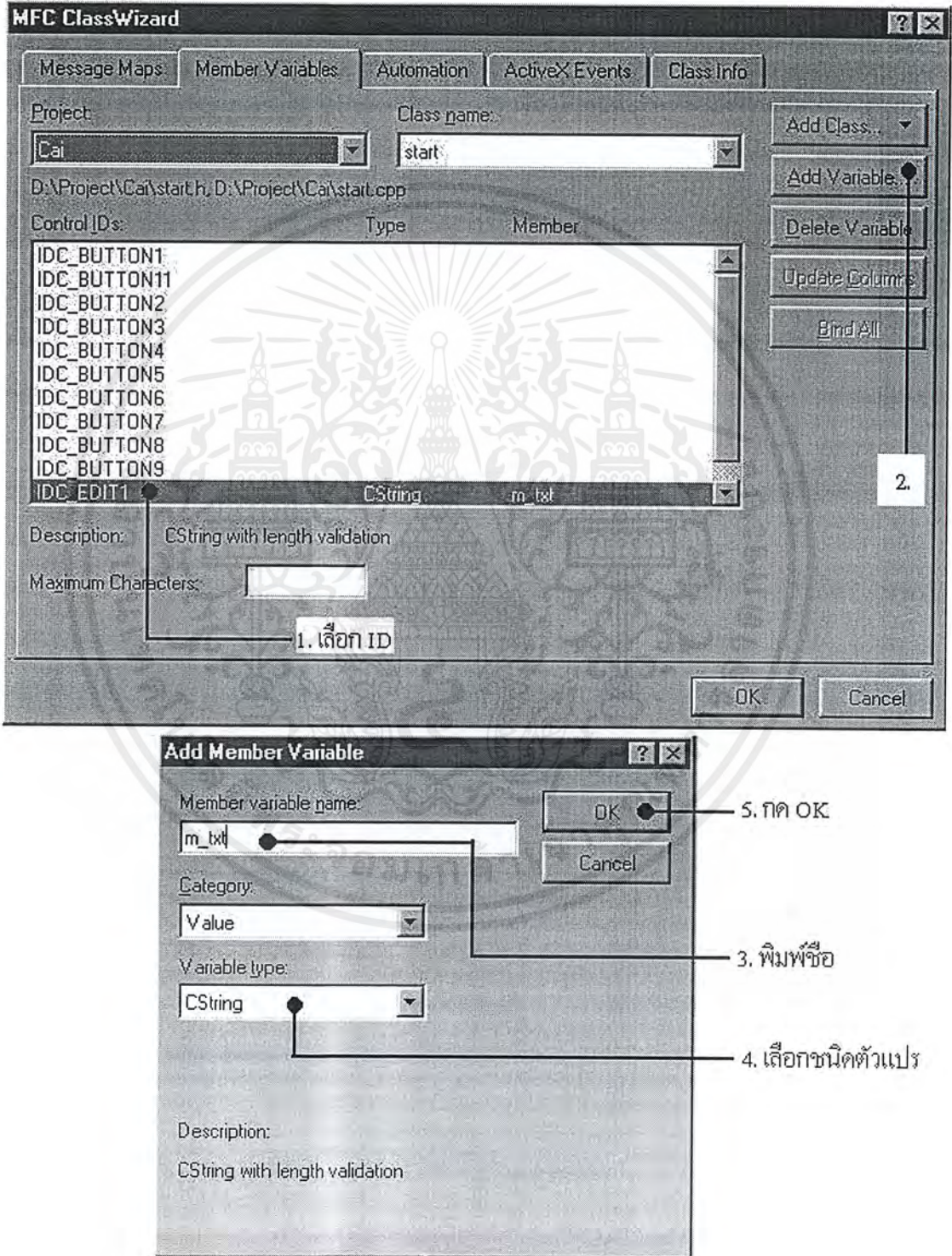
    (“คลาส”,WS_CHILD|WS_VISIBLE|BS_PUSHBUTTON,crect(530,425,695,455),this,
    IDC_TT4);

```

รูปที่ 3.11 โปรแกรมสร้างปุ่มกด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโปรแกรมในรูปที่ 3.12 ยังไม่สามารถ Run ได้เนื่องจากมีการใช้ตัวแปรชนิด DDX ดังนั้นจะต้องทำการเพิ่มตัวแปรชนิดนี้ให้กับ Text Box โดยใช้ ClassWizard ดังรูปที่ 3.13



รูปที่ 3.13 การสร้างตัวแปร DDX ให้กับ Text Box

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลเนื้อหาไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

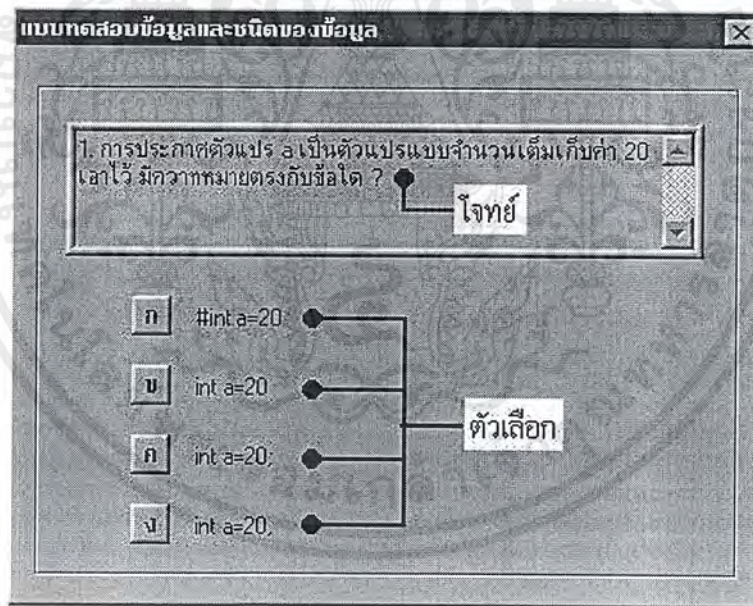
เมื่อทำการสร้างเมนูย่อยแล้วจะต้องทำการเม็ปแมสเสจ ให้กับปุ่มกดที่สร้างขึ้นใหม่ เพื่อทำการเรียกบทเรียนขึ้นมา การเรียกโปรแกรม ScreenCam เพื่อทำการเปิดไฟล์บทเรียนนั้นจะใช้ ฟังก์ชัน ShellExecute() ดังรูปที่ 3.14

```
ShellExecute(NULL,"open","Scplayer.exe ", "mov5_4.scm", NULL, SW_SHOWNORMAL);
```

รูปที่ 3.14 การใช้ฟังก์ชัน ShellExecute()

3.8.2 การสร้างแบบทดสอบ

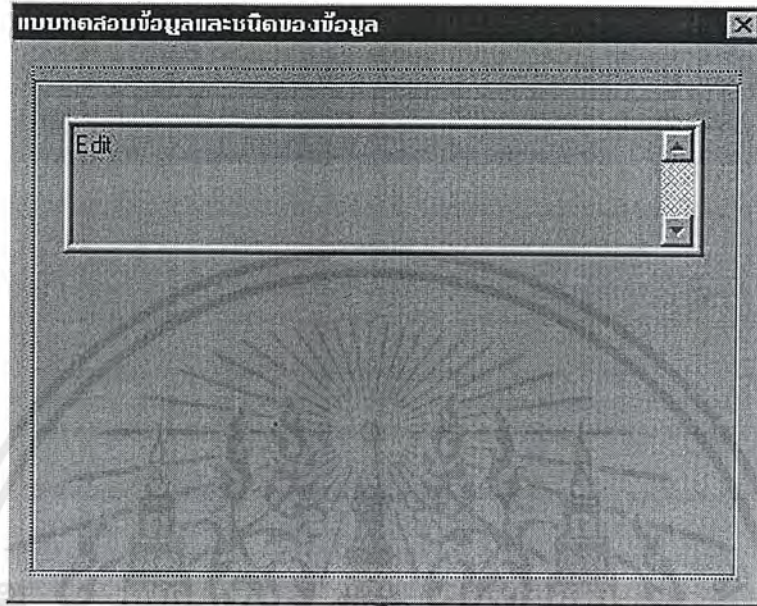
การสร้างแบบทดสอบท้ายบทเรียน ได้ออกแบบไว้ดังรูปที่ 3.15



รูปที่ 3.15 รูปแบบของแบบทดสอบที่ได้ออกแบบไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.16 เป็นไดอะล็อกที่สร้างไว้

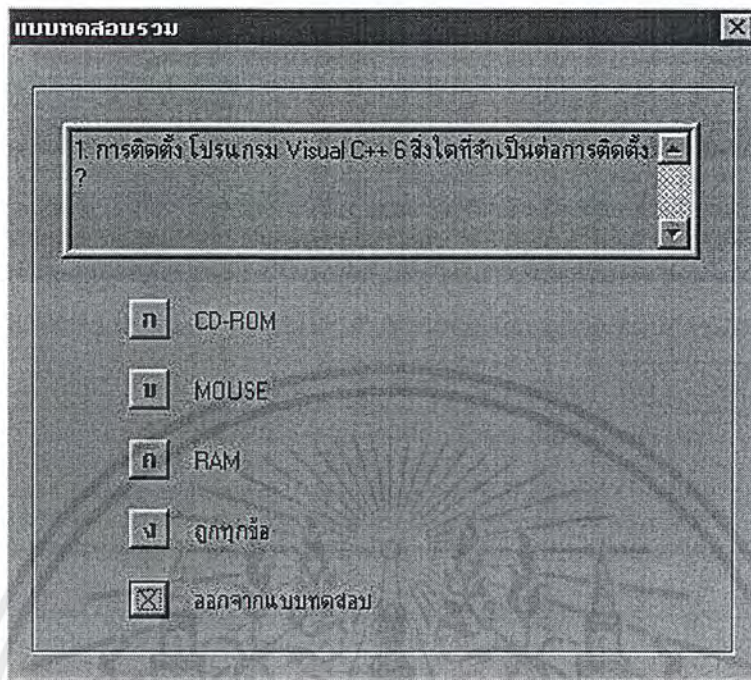


รูปที่ 3.16 ไดอะล็อกของข้อสอบที่สร้าง

3.8.3 การสร้างแบบทดสอบรวม

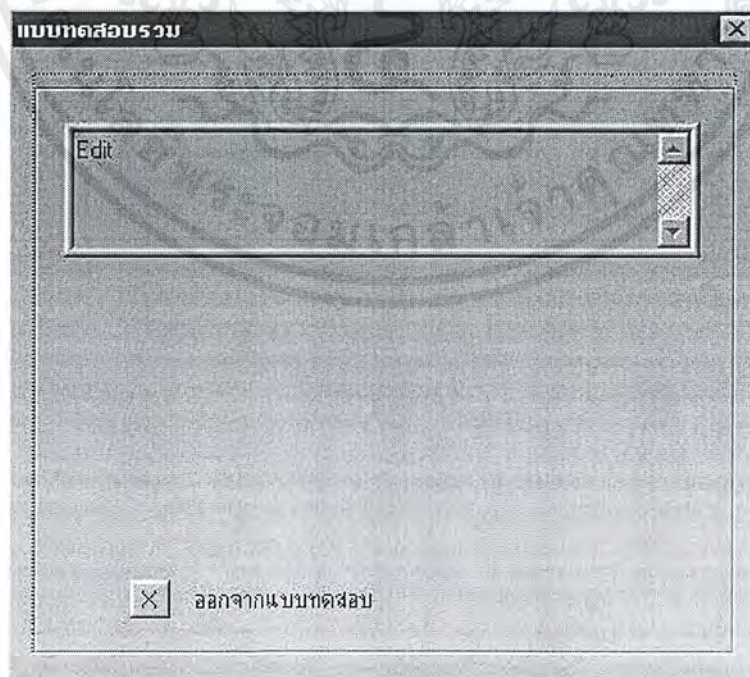
การสร้างแบบทดสอบรวมได้ออกแบบไว้ดังรูปที่ 3.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



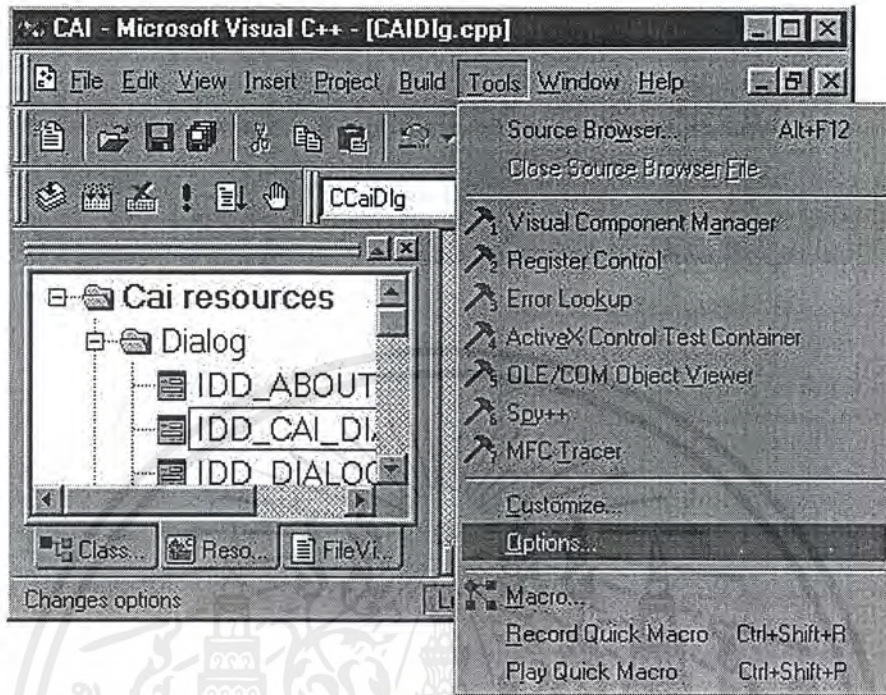
รูปที่ 3.17 แบบทดสอบรวมที่ได้ออกแบบ

รูปที่ 3.18 เป็นไอคอนแบบทดสอบรวมที่ได้สร้างไว้



รูปที่ 3.18 ไอคอนของแบบทดสอบรวม

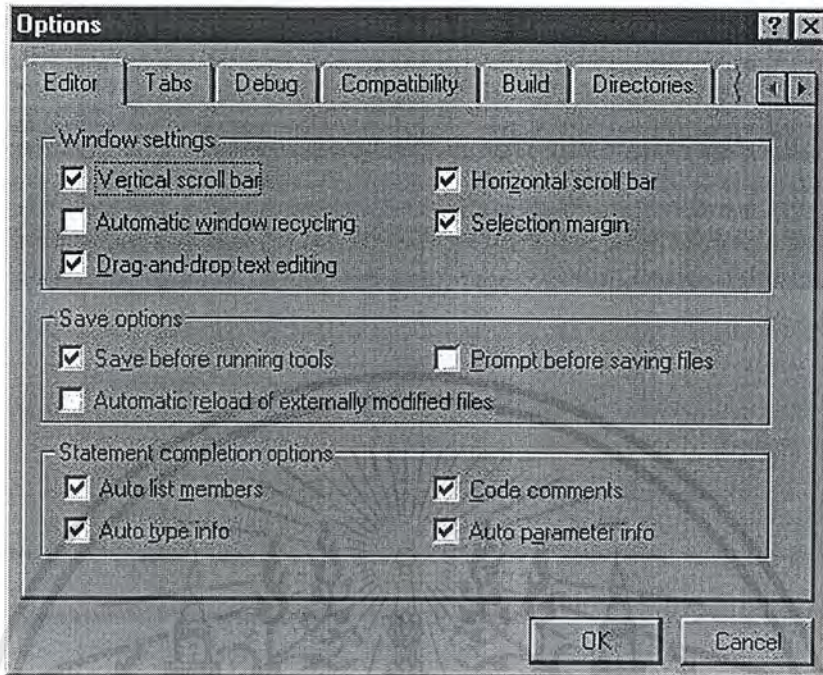
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



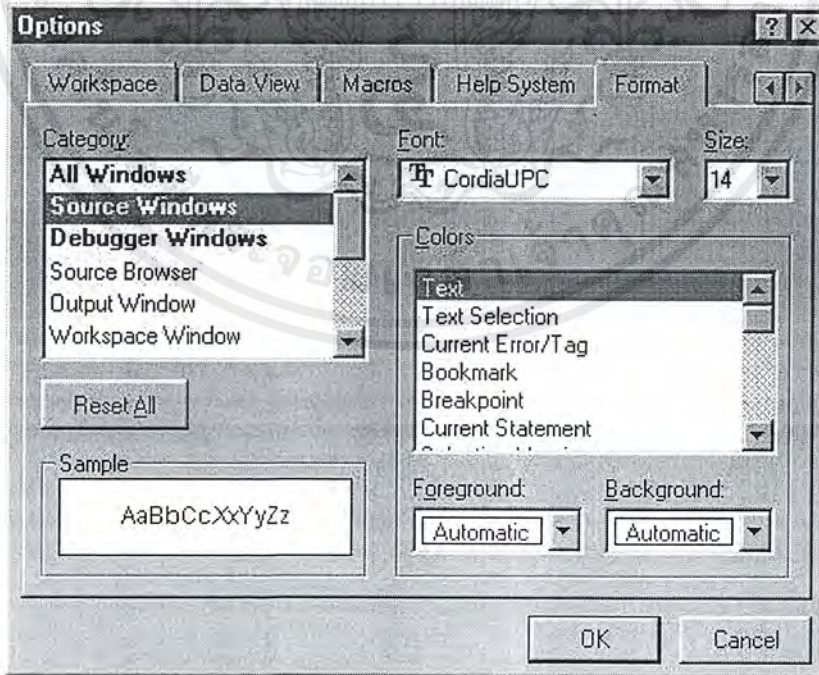
รูปที่ 3.20 การเรียกใช้ออปชันของ โปรแกรม Visual C++ 6.0

หลังจากนั้นจะพบไดอะล็อก Options ดังรูปที่ 3.21 หลังจากนั้นทำการเลื่อนไปที่แท็บ Format ซึ่งอยู่ที่ท้ายสุดของไดอะล็อก

ที่ Category เลือก Source Windows และที่ Font เลือกรูปแบบของตัวอักษรที่สามารถแสดงภาษาไทยได้ พร้อมทั้งปรับขนาดให้เหมาะสมต่อการใช้งาน ดังรูปที่ 3.22



รูปที่ 3.21 ไอคอน Options

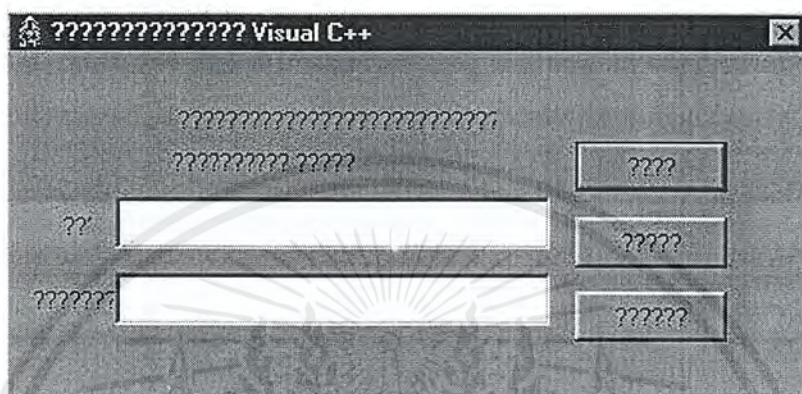


รูปที่ 3.22 การปรับแต่งรูปแบบตัวอักษรให้สามารถแสดงภาษาไทยได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

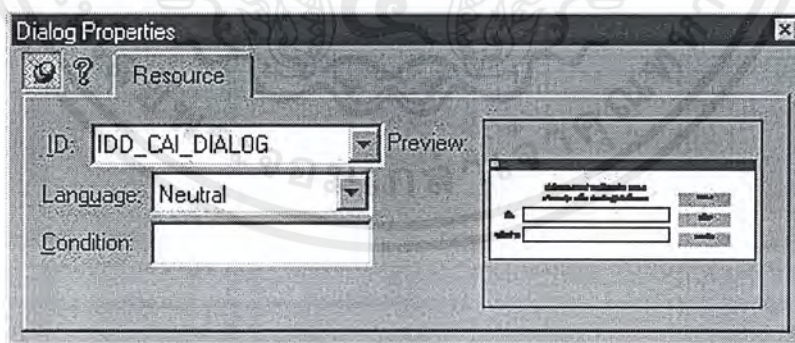
3.8.5 การสร้างแอปพลิเคชันภาษาไทย

โปรเจกต์ที่ทำการสร้างขึ้นนี้ มีการแสดงผลเป็นภาษาไทย แต่ยังไม่สามารถแสดงภาษาไทย ออกมาได้เมื่อทำการรัน โปรแกรม ดังรูปที่ 3.23



รูปที่ 3.23 ปัญหาในการแสดงผลเป็นภาษาไทย

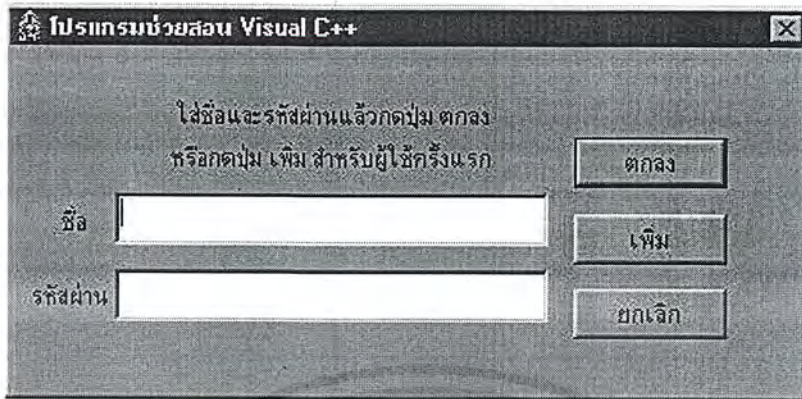
ดังนั้นจึงต้องทำการปรับเปลี่ยนคุณสมบัติของไดอะล็อกโดยใช้ Dialog Properties ในการแก้ไขการแสดงผลภาษาให้สามารถแสดงภาษาไทยได้นั้น ให้เปลี่ยน Language เป็น Neutral ดังรูปที่ 3.24



รูปที่ 3.24 การแก้ไขภาษา

เมื่อทำการรัน โปรแกรมอีกครั้งจะเป็นดังรูปที่ 3.25

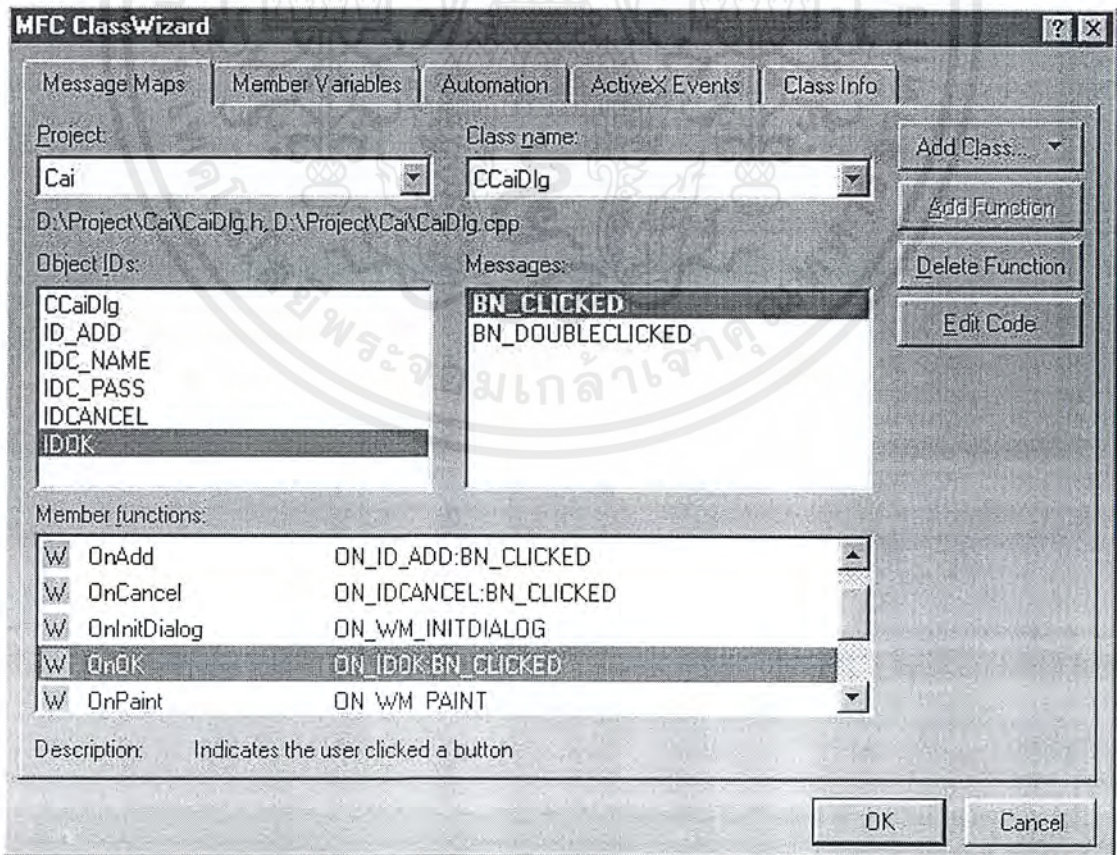
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.25 ผลการรันเมื่อเปลี่ยนภาษาที่ใช้แสดง

3.8.6 การแมปเมสเสจให้โครงการ

การแมปเมสเสจให้โครงการนี้สามารถทำได้โดยใช้ ClassWizard ในการแมปเมสเสจ ดังรูปที่ 3.26



เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ 3.26 การแมปเมสเสจให้กับปุ่มกดที่ใช้ในโครงการนี้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

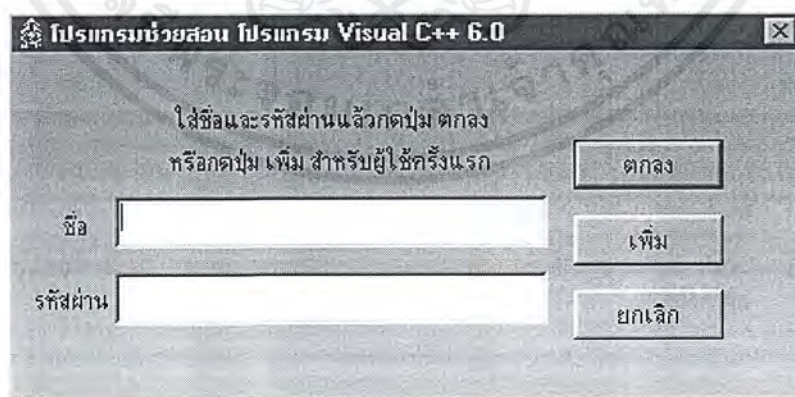
การทดลองและผลการทดลอง

4.1 ความต้องการของโปรแกรม

1. เครื่องคอมพิวเตอร์ Pentium 100 ขึ้นไป
2. หน่วยความจำ 16 Mbytes ขึ้นไป
3. พื้นที่ว่างในฮาร์ดดิสก์ 10 Mbytes ขึ้นไป
4. เม้าส์และคีย์บอร์ด
5. ระบบปฏิบัติการ Windows 95 หรือ Windows 98
6. Sound Card
7. CD – ROM Drive

4.2 การเข้าสู่โปรแกรม

โปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0 เมื่อนำแผ่น CD – ROM โปรแกรมใส่เข้าไปใน CD – ROM Drive โปรแกรมช่วยสอนจะเริ่มทำงานทันที เมื่อโปรแกรมเริ่มทำงาน จะปรากฏไดอะล็อกให้ทำการลงทะเบียนดังรูปที่ 4.1



โปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0

ใส่ชื่อและรหัสผ่านแล้วกดปุ่ม ตกลง
หรือคลิกปุ่ม เพิ่ม สำหรับผู้ใช้ครั้งแรก

ชื่อ

รหัสผ่าน

ตกลง

เพิ่ม

ยกเลิก

รูปที่ 4.1 การลงทะเบียน

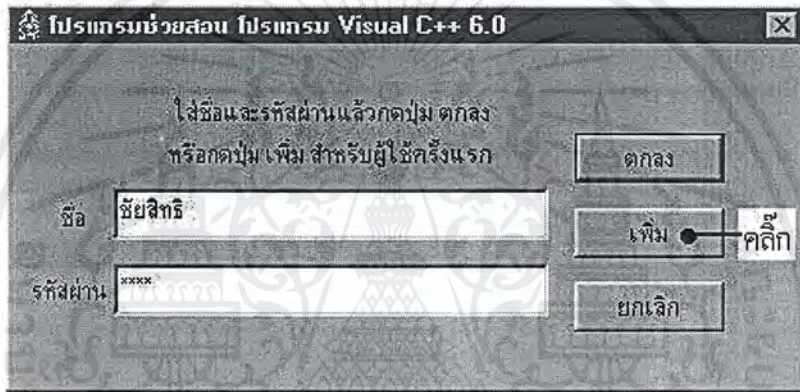
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การลงทะเบียนเพื่อทำการเข้าสู่บทเรียน

ก่อนที่จะทำการเข้าสู่บทเรียน จะต้องผ่านการลงทะเบียนเสียก่อน โดยการลงทะเบียนนี้ แบ่งออกเป็น 2 กรณีคือ

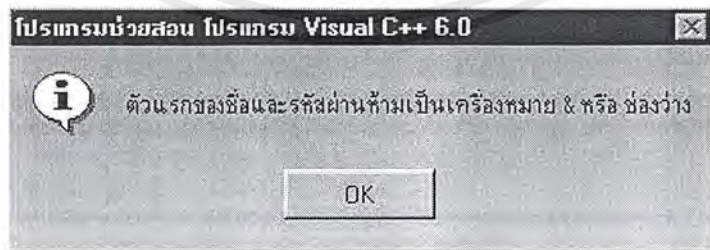
4.3.1 ผู้ใช้ยังไม่เคยลงทะเบียนมาก่อน

ถ้าหากผู้ใช้ยังไม่เคยผ่านการลงทะเบียนมาก่อน จะต้องทำการลงทะเบียนก่อน โดยใส่ชื่อลงในช่อง “ชื่อ” และใส่รหัสผ่านลงในช่อง “รหัสผ่าน” หลังจากนั้นให้คลิกเมาส์ที่ปุ่ม “เพิ่ม” ดังรูปที่ 4.2



รูปที่ 4.2 การลงทะเบียนครั้งแรก

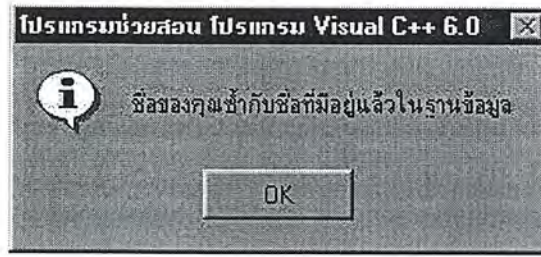
ในการใส่ชื่อและรหัสนั้นตัวแรกจะต้องไม่เป็นเครื่องหมาย “&” หากใส่ชื่อและรหัสผ่านไม่ถูกต้องจะพบไดอะล็อกเตือนดังรูปที่ 4.3



รูปที่ 4.3 ไดอะล็อกเตือนการใส่ชื่อและรหัสผ่าน

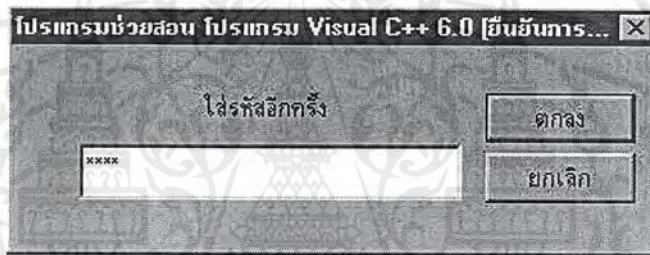
ถ้าหากชื่อที่ใส่นั้น ซ้ำกับชื่อที่มีอยู่แล้วในระบบจะพบไดอะล็อกเตือนดังรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



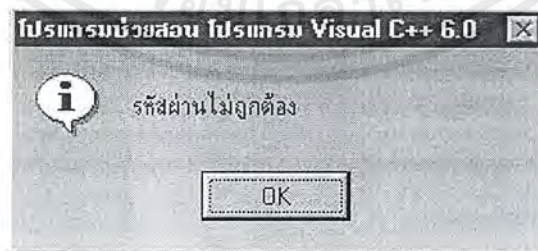
รูปที่ 4.4 ไดอะล็อกเตือนการใส่ชื่อซ้ำกับที่มีอยู่แล้วในระบบ

หลังจากนั้นจะพบกับ ไดอะล็อกยืนยันรหัสผ่าน ดังรูปที่ 4.5



รูปที่ 4.5 การยืนยันรหัสผ่าน

ถ้าใส่รหัสไม่ตรงกับครั้งแรก จะพบไดอะล็อกเตือนการใส่รหัส และจะให้ทำการใส่รหัสอีกครั้ง ดังรูปที่ 4.6



รูปที่ 4.6 ไดอะล็อกเตือนการใส่รหัส

เมื่อใส่รหัสถูกต้องแล้วจะพบกับไดอะล็อกแสดงคำแนะนำโปรแกรม ดังรูปที่ 4.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

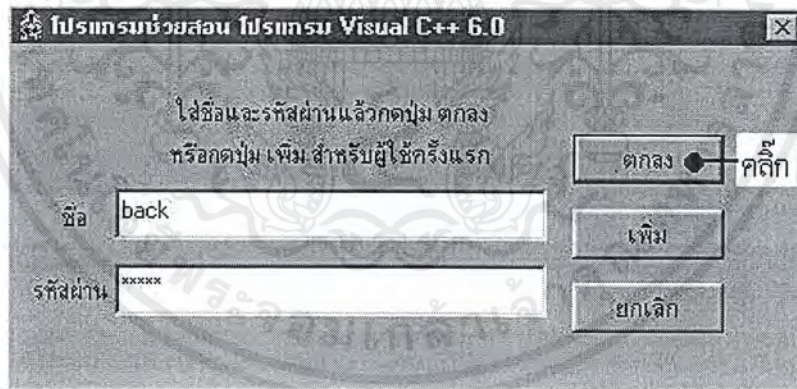


รูปที่ 4.7 คำแนะนำของ โปรแกรม

หลังจากนั้นจะเข้าสู่เมนูหลัก

4.3.2 ผู้ใช้ผ่านลงทะเบียนมาก่อนแล้ว

สำหรับผู้ที่ใช้ผ่านการลงทะเบียนมาก่อนแล้ว ก็จะทำการใส่ชื่อและรหัสผ่าน หลังจากนั้น จึงคลิกเมาส์ที่ปุ่ม “ตกลง” ดังรูปที่ 4.8 หลังจากนั้นจะเข้าสู่เมนูหลัก

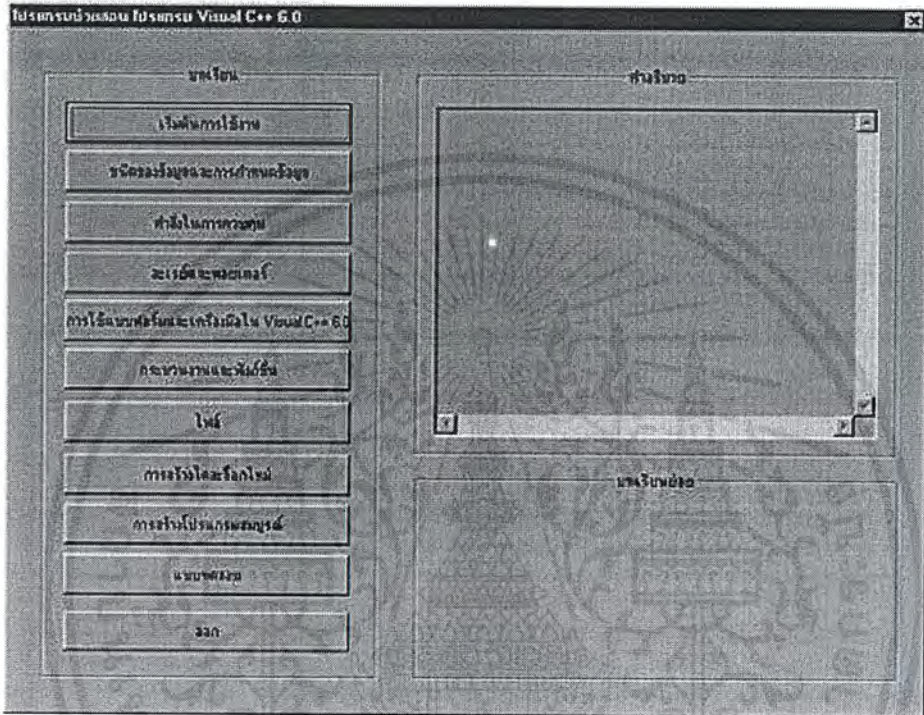


รูปที่ 4.8 การเข้าสู่บทเรียนของผู้ที่ผ่านการลงทะเบียนมาแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 เมนูหลัก

เมื่อผ่านการลงทะเบียนเรียบร้อยแล้ว โปรแกรมจะเข้าสู่เมนูหลักดังรูปที่ 4.9



รูปที่ 4.9 เมนูหลัก

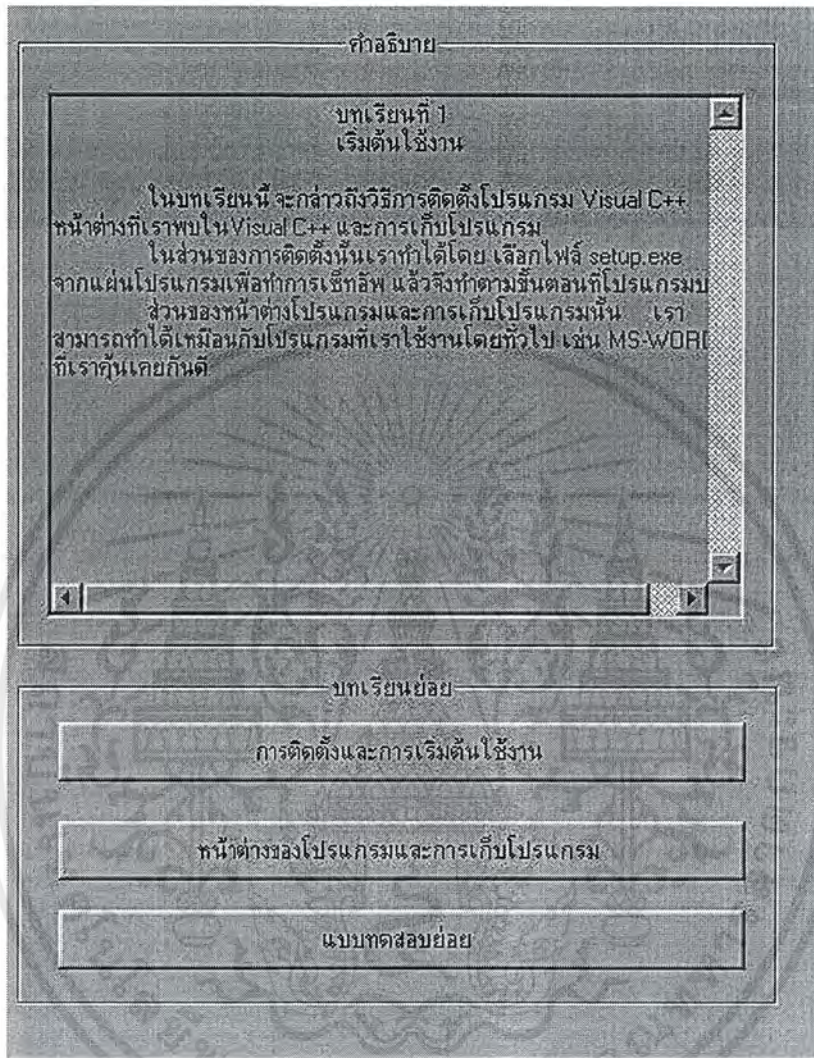
เมื่อมีการคลิกที่เมนูหลัก จะมีเมนูย่อยแสดงออกมา และในเท็กซ์บ็อกซ์ จะแสดงรายละเอียดในหัวข้อหลักนั้น

4.5 การเลือกบทเรียน

การเลือกบทเรียนนั้นสามารถทำได้โดยเมื่อเลือกที่เมนูหลัก จะแสดงเมนูย่อยออกมาพร้อมทั้งรายละเอียดในบทเรียนนั้น ๆ

การเลือกบทเรียนที่ 1 สามารถทำได้โดยคลิกเมาส์ที่ปุ่ม “เริ่มต้นการใช้งาน” ในเมนูหลัก จะแสดงรายละเอียด และเมนูย่อยของบทเรียนที่ 1 ดังรูปที่ 4.10

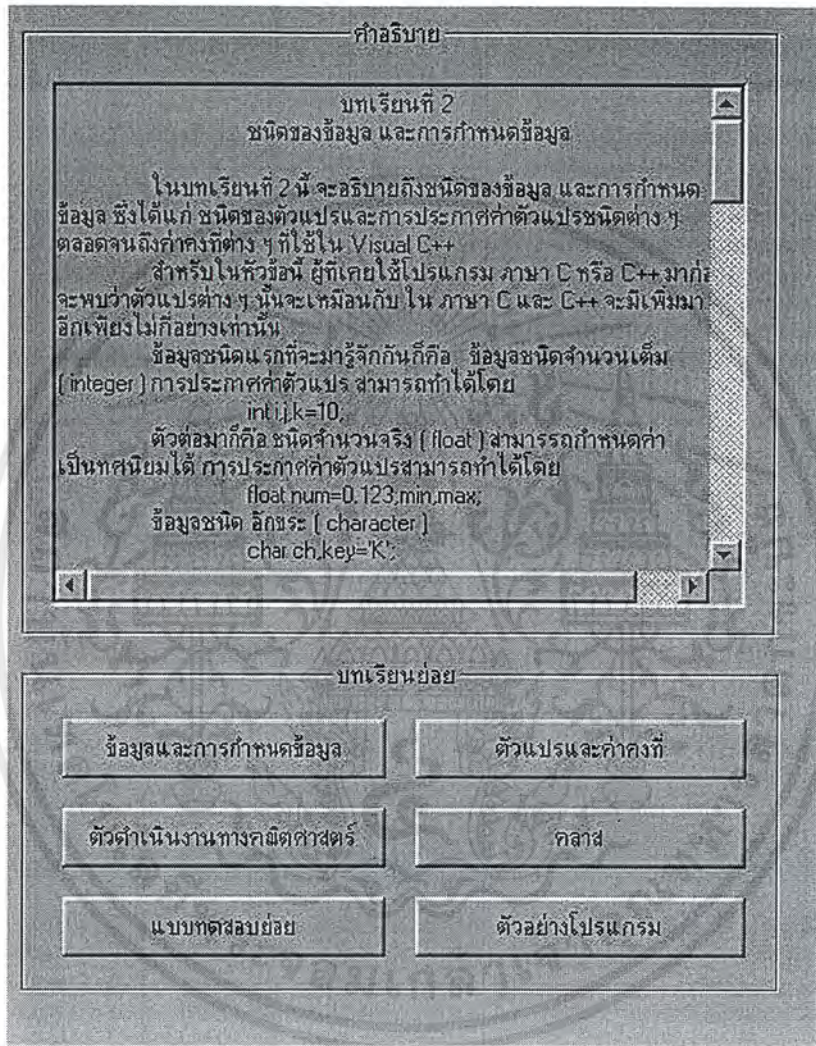
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 รายละเอียดและเมนูย่อยของบทเรียนที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

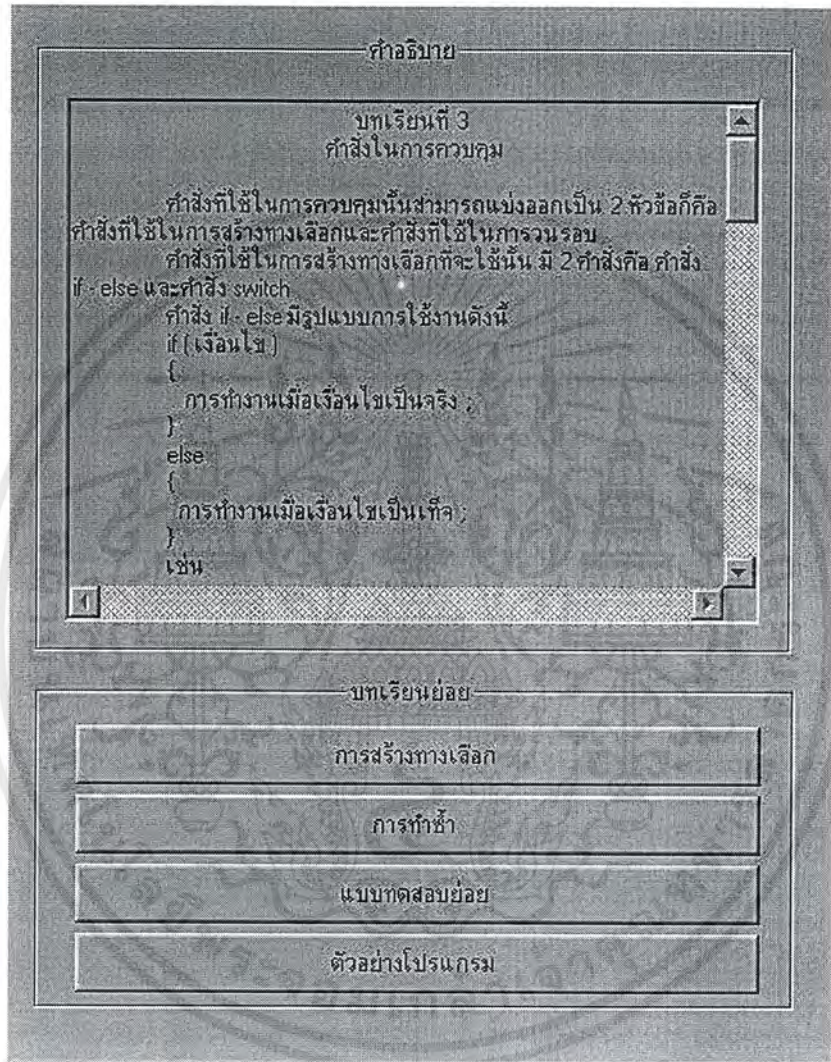
การเลือกบทเรียนที่ 2 สามารถทำได้โดยคลิกเมาส์ที่ปุ่ม “ชนิดของข้อมูล และการกำหนดข้อมูล” ในเมนูหลักจะแสดงรายละเอียด และเมนูย่อยของบทเรียนที่ 2 ดังรูปที่ 4.11



รูปที่ 4.11 รายละเอียดและเมนูย่อยของบทเรียนที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

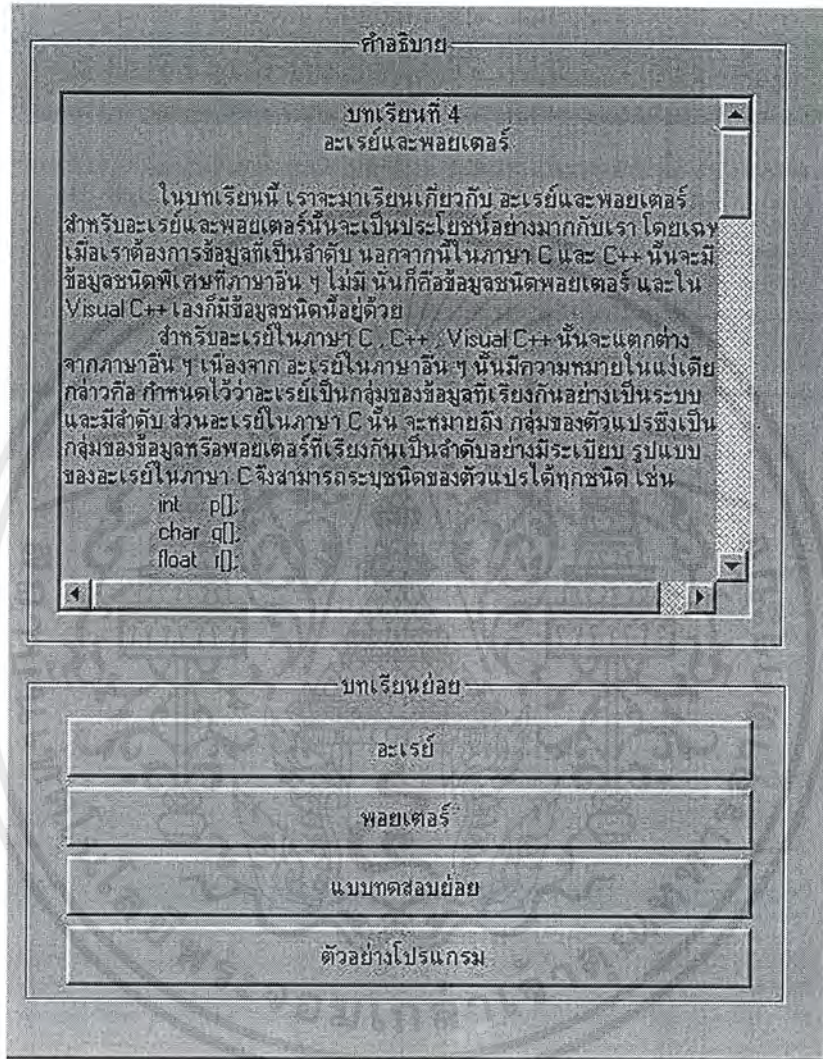
การเลือกบทเรียนที่ 3 สามารถทำได้โดยคลิกเมาส์ที่ปุ่ม “คำสั่งในการควบคุม” ในเมนูหลัก จะแสดงรายละเอียด และเมนูย่อยของบทเรียนที่ 3 ดังรูปที่ 4.12



รูปที่ 4.12 รายละเอียดและเมนูย่อยของบทเรียนที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

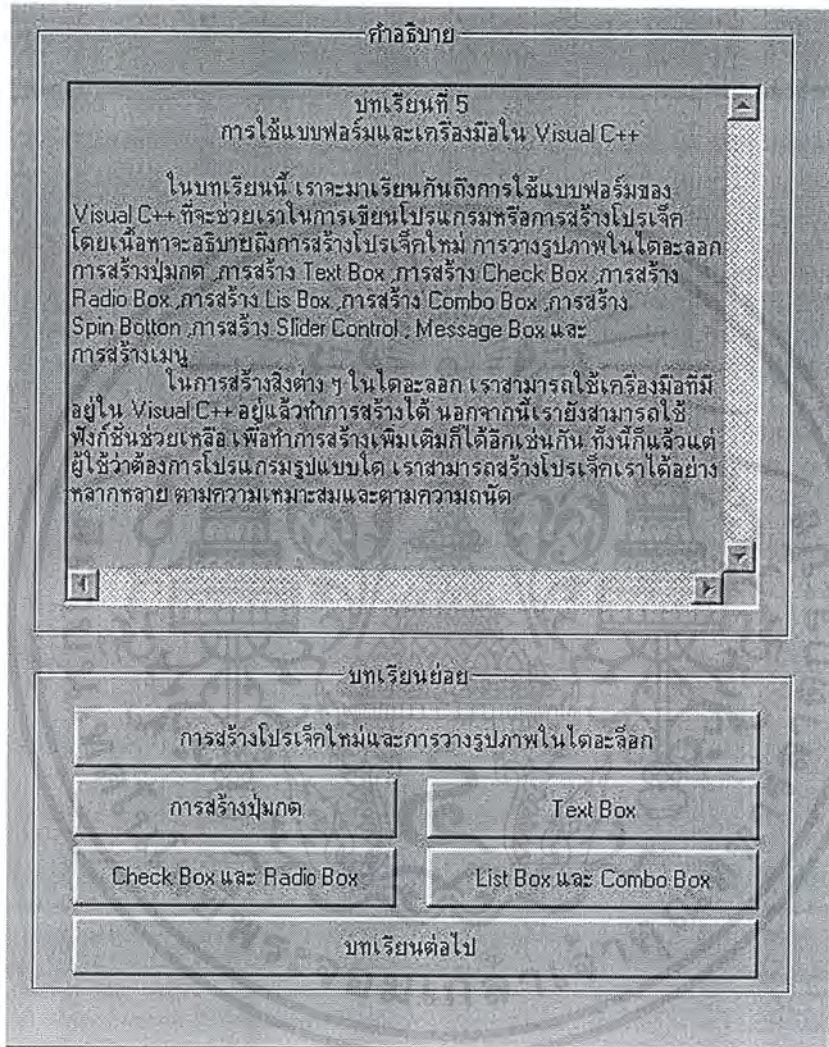
การเลือกบทเรียนที่ 4 สามารถทำได้โดยคลิกเมาส์ที่ปุ่ม “อะเรย์และพอยเตอร์” ในเมนูหลัก จะแสดงรายละเอียด และเมนูย่อยของบทเรียนที่ 4 ดังรูปที่ 4.13



รูปที่ 4.13 รายละเอียดและเมนูย่อยของบทเรียนที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

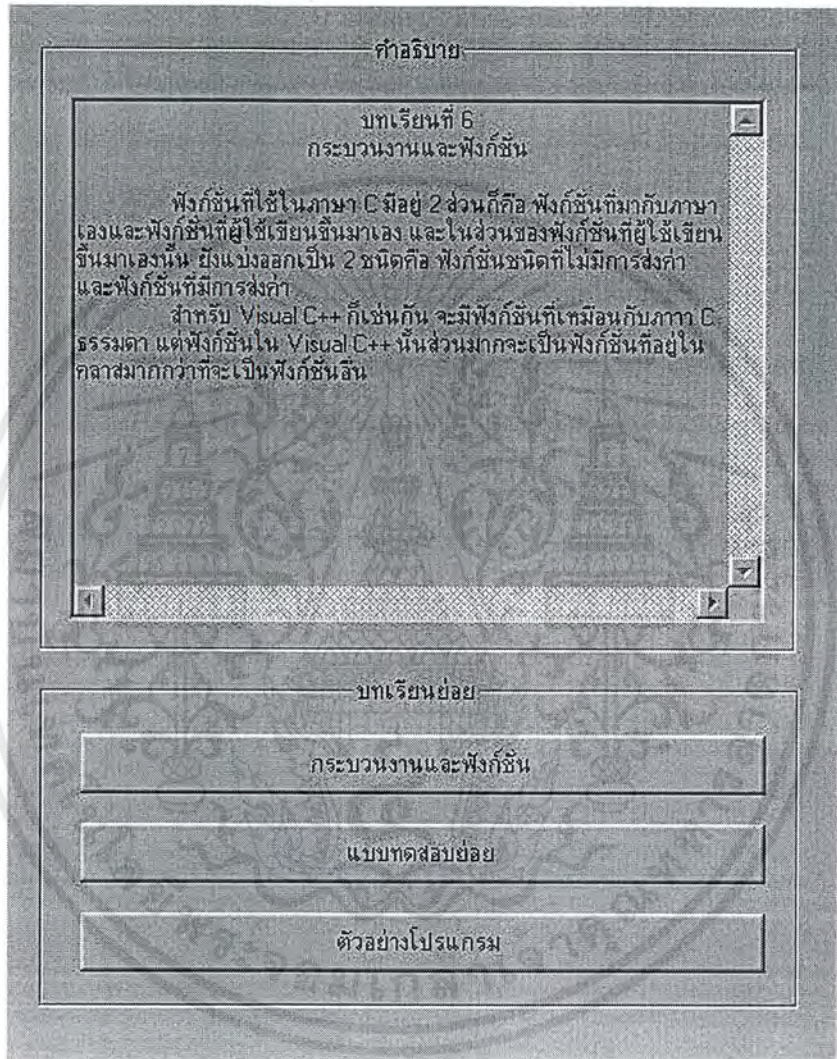
การเลือกบทเรียนที่ 5 สามารถทำได้โดยคลิกเมาส์ที่ปุ่ม “การใช้แบบฟอร์มและเครื่องมือใน Visual C++ 6.0” ในเมนูหลัก จะแสดงรายละเอียด และเมนูย่อยของบทเรียนที่ 5 ดังรูปที่ 4.14



รูปที่ 4.14 รายละเอียดและเมนูย่อยของบทเรียนที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

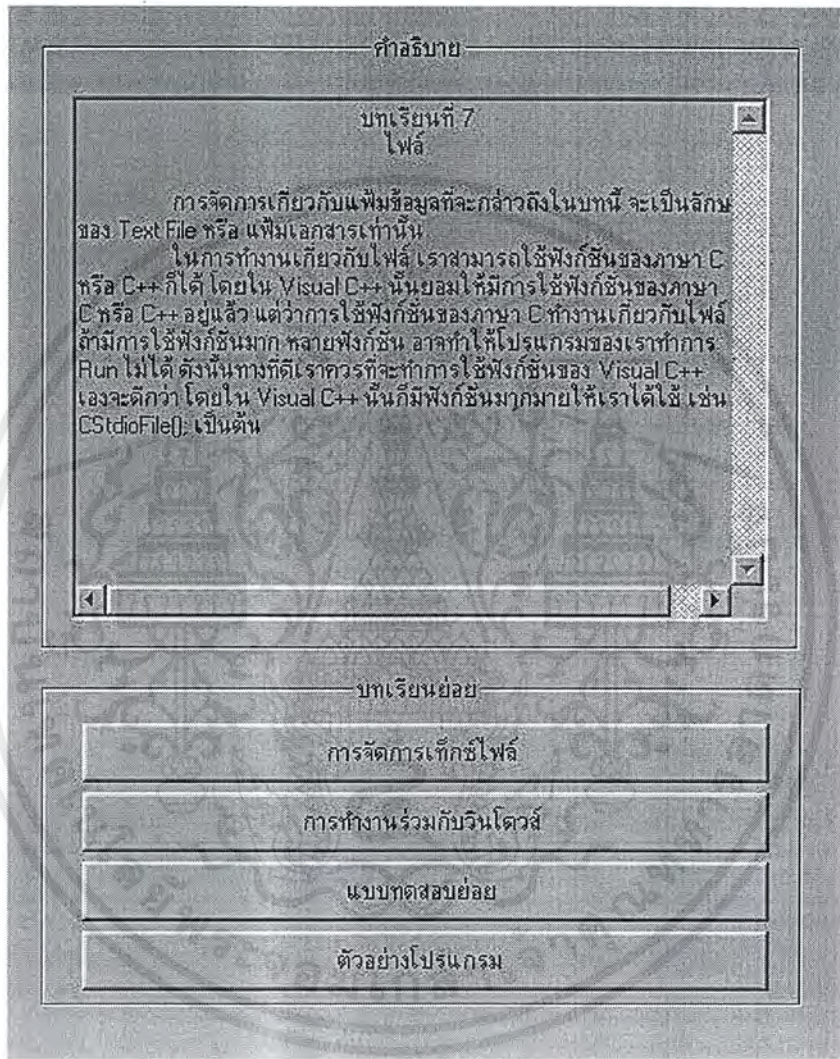
การเลือกบทเรียนที่ 6 สามารถทำได้โดยคลิกเมาส์ที่ปุ่ม “กระบวนงานและฟังก์ชัน” ในเมนูหลัก จะแสดงรายละเอียด และเมนูย่อยของบทเรียนที่ 6 ดังรูปที่ 4.15



รูปที่ 4.15 รายละเอียดและเมนูย่อยของบทเรียนที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

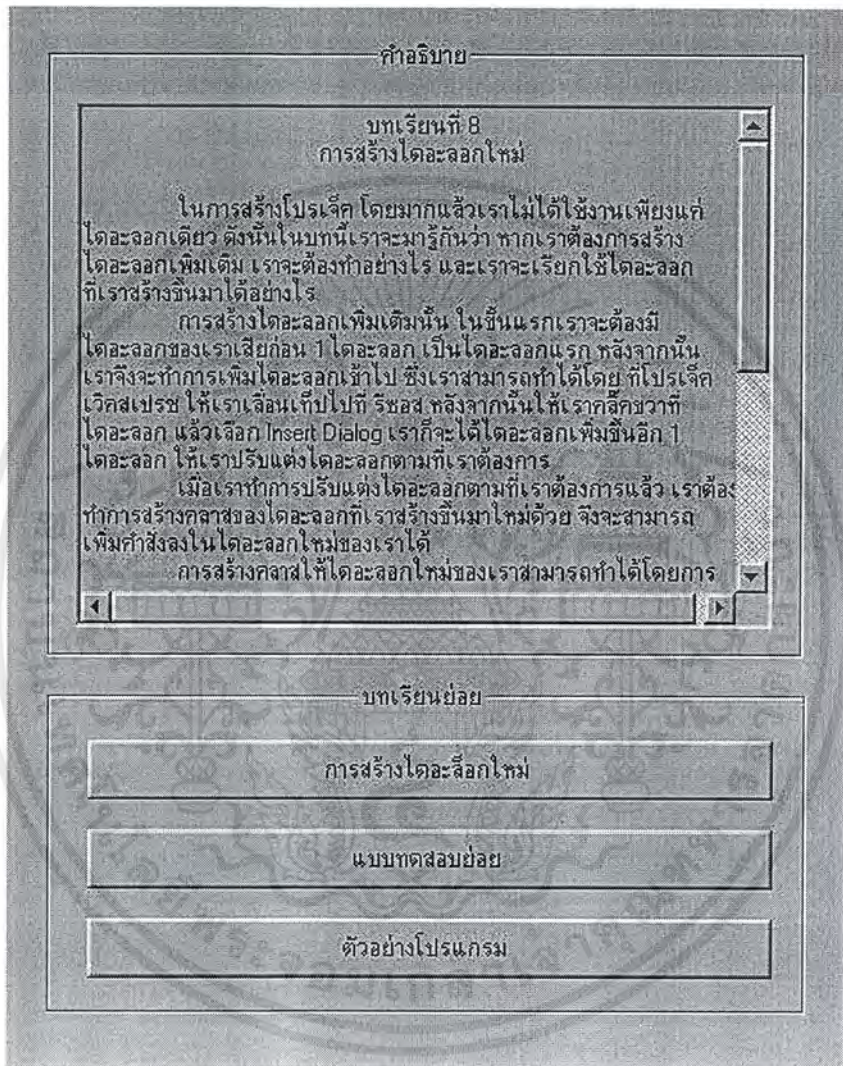
การเลือกบทเรียนที่ 7 สามารถทำได้โดยคลิกเมาส์ที่ปุ่ม “ไฟล์” ในเมนูหลัก จะแสดงรายละเอียด และเมนูย่อยของบทเรียนที่ 7 ดังรูปที่ 4.16



รูปที่ 4.16 รายละเอียดและเมนูย่อยของบทเรียนที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

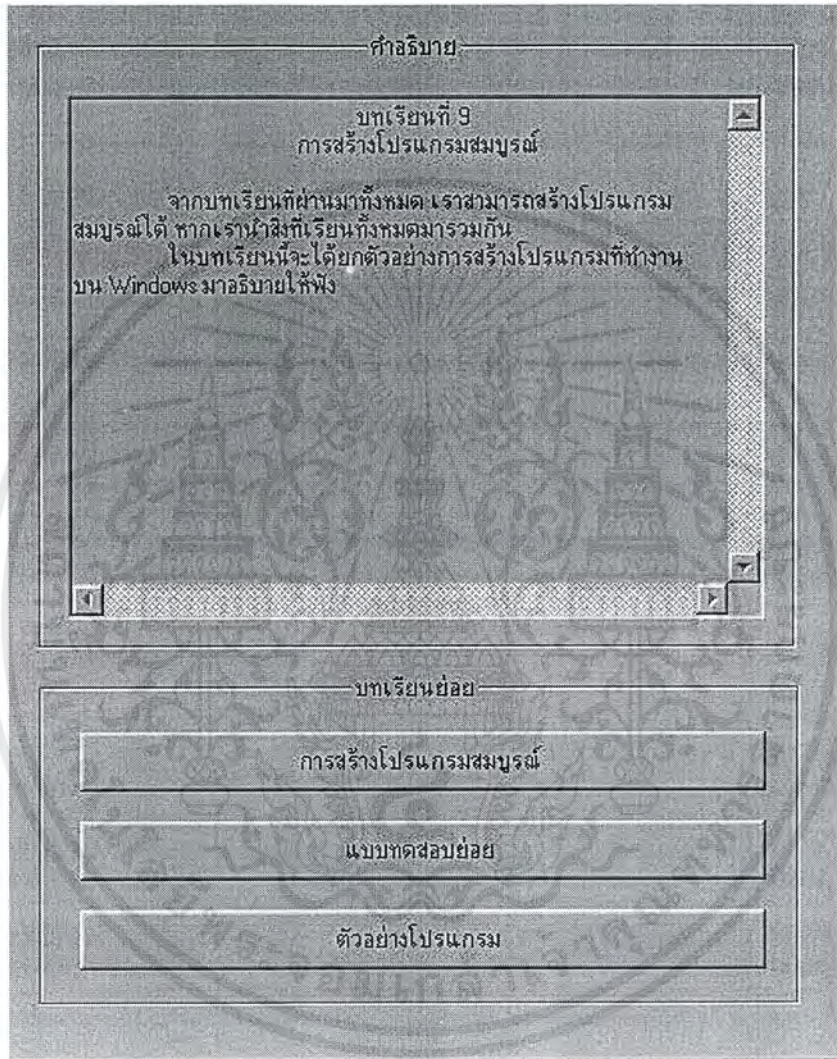
การเลือกบทเรียนที่ 8 สามารถทำได้โดยคลิกเมาส์ที่ปุ่ม “การสร้างไดอะล็อกใหม่” ในเมนูหลัก จะแสดงรายละเอียด และเมนูย่อยของบทเรียนที่ 8 ดังรูปที่ 4.17



รูปที่ 4.17 รายละเอียดและเมนูย่อยของบทเรียนที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเลือกบทเรียนที่ 9 สามารถทำได้โดยคลิกเมาส์ที่ปุ่ม “ชนิดของข้อมูลและการกำหนดข้อมูล” ในเมนูหลัก จะแสดงรายละเอียด และเมนูย่อยของบทเรียนที่ 9 ดังรูปที่ 4.18



รูปที่ 4.18 รายละเอียดและเมนูย่อยของบทเรียนที่ 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 การเข้าสู่บทเรียน

เมื่อจะเข้าสู่บทเรียนสามารถทำได้โดยการคลิกเมาส์ที่ปุ่มภายในเมนูย่อยของบทเรียนนั้น ๆ โปรแกรมจะทำการเรียกโปรแกรม ScreenCam ขึ้นมารัน พร้อมทั้งทำการเปิดไฟล์บทเรียน

4.7 การคัดลอกตัวอย่างโปรแกรม

เมื่อจะทำการคัดลอกตัวอย่างโปรแกรม ให้คลิกเมาส์ที่ปุ่มตัวอย่างของเมนูย่อยของบทเรียนนั้นๆ เมื่อโปรแกรมคัดลอกตัวอย่างโปรแกรมเรียบร้อยแล้ว จะพบกับไดอะล็อกคังรูปที่ 4.19



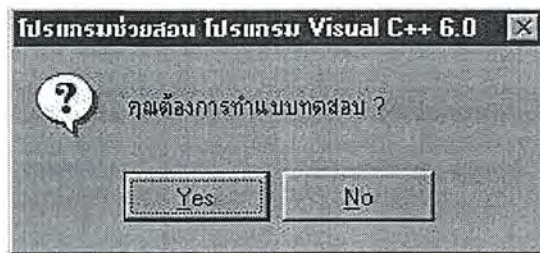
รูปที่ 4.19 เมื่อ โปรแกรมคัดลอกไฟล์ตัวอย่างเสร็จสมบูรณ์

ตัวอย่างโปรแกรมจะถูกคัดลอกไปที่ C:\Sample โดยไฟล์ตัวอย่างนี้จะเป็นไฟล์ที่มีนามสกุลเป็น .ZIP ผู้ใช้จะต้องทำการ Unzip เสียก่อน ซึ่งผู้ใช้อาจใช้โปรแกรม WinZip ก็ได้

4.8 แบบทดสอบท้ายบทเรียน

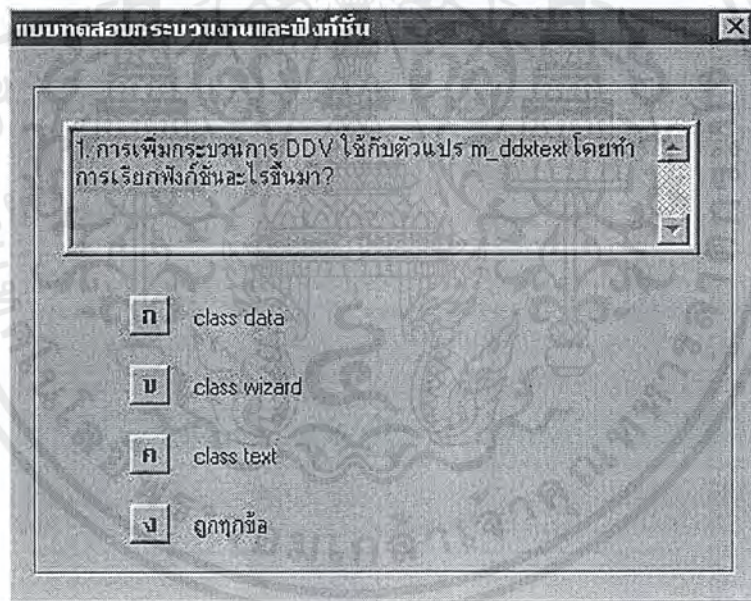
เมื่อผู้เรียน เรียนจบบทเรียนในแต่ละบท ผู้เรียนสามารถทดสอบตนเองได้โดยทำแบบทดสอบท้ายบทเรียน ซึ่งอยู่ภายในเมนูย่อยของบทเรียนนั้น ๆ การเข้าสู่แบบทดสอบท้ายบทเรียนนั้นสามารถทำได้โดยคลิกเมาส์ที่ปุ่ม “แบบทดสอบย่อย” ในเมนูย่อยของในแต่ละบทเรียน

เมื่อผู้เรียนคลิกเมาส์ที่แบบทดสอบท้ายบทเรียนจะพบกับ ไดอะล็อกคังรูปที่ 4.20



รูปที่ 4.20 ไอคอนสอบถามความต้องการในการทำแบบทดสอบ

เมื่อคลิกปุ่ม “Yes” ก็จะเข้าสู่แบบทดสอบ แต่ถ้าคลิกปุ่ม “No” ก็จะกลับสู่เมนูหลัก เมื่อเข้าสู่แบบทดสอบท้ายบทเรียนจะพบกับ ไอคอน ดังรูปที่ 4.21



รูปที่ 4.21 ไอคอนแบบทดสอบ

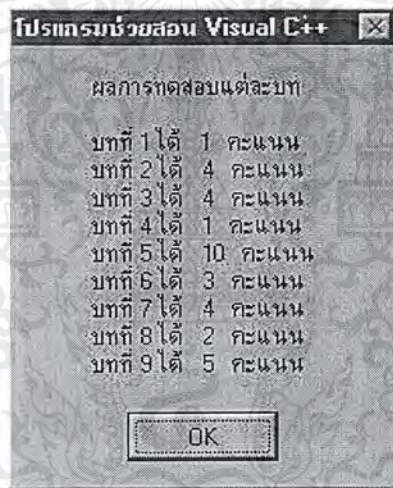
ผู้เรียนจะต้องทำแบบทดสอบจนครบทุกข้อ จึงจะออกจากแบบทดสอบได้ หรือคลิกที่ปุ่ม ปิดโปรแกรม ก็จะสามารถออกจากโปรแกรมได้เช่นกันและเมื่อทำข้อสอบข้อใดไปแล้วไม่สามารถกลับมาแก้ไขได้อีก เมื่อทำข้อสอบจนเสร็จแล้ว จะพบไอคอนในการแสดงผลการสอบในครั้งนี ดังรูปที่ 4.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



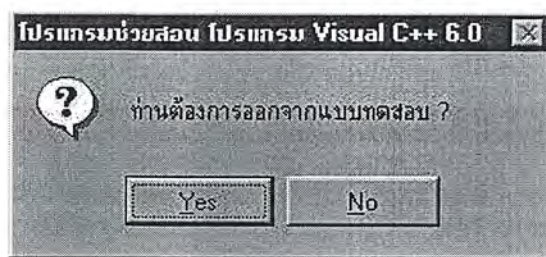
รูปที่ 4.22 ผลการทดสอบท้ายบทเรียน

และจะพบกับ ไดอะล็อกแสดงคะแนนสอบในแต่ละบทดังรูปที่ 4.23



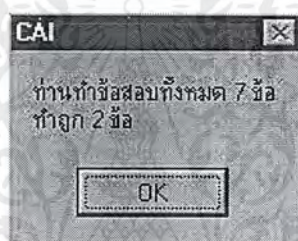
รูปที่ 4.23 คะแนนสอบในแต่ละบท

เมื่อต้องการออกจากแบบทดสอบก่อนที่จะทำข้อสอบจนครบสามารถทำได้โดยการคลิกที่ปุ่ม Close ที่บริเวณมุมบนขวา เมื่อคลิกแล้วจะพบกับ ไดอะล็อกดังรูปที่ 4.24



รูปที่ 4.24 ไอคอนถามความต้องการในการออกจากแบบทดสอบ

เมื่อคลิกที่ปุ่ม Yes แล้วจะพบกับ ไอคอนดังรูปที่ 4.25



รูปที่ 4.25 ไอคอนรายงานผลการทำข้อสอบ

4.9 แบบทดสอบรวม

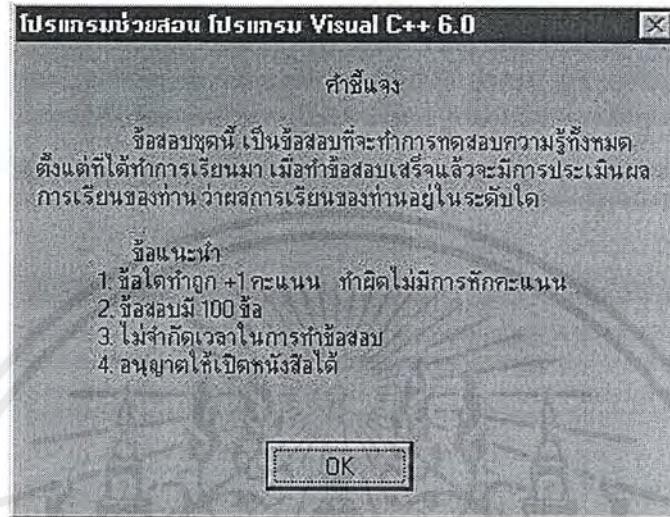
เมื่อผู้เรียนต้องการทำแบบทดสอบรวมเมื่อผ่านการเรียนครบทุกบทแล้ว สามารถทำได้โดยการคลิกเมาส์ที่แบบทดสอบในเมนูหลัก เมื่อคลิกเมาส์แล้วจะพบกับไอคอนเพื่อถามความต้องการในการทำแบบทดสอบ ดังรูปที่ 4.26



รูปที่ 4.26 ไอคอนถามความต้องการในการทำแบบทดสอบ

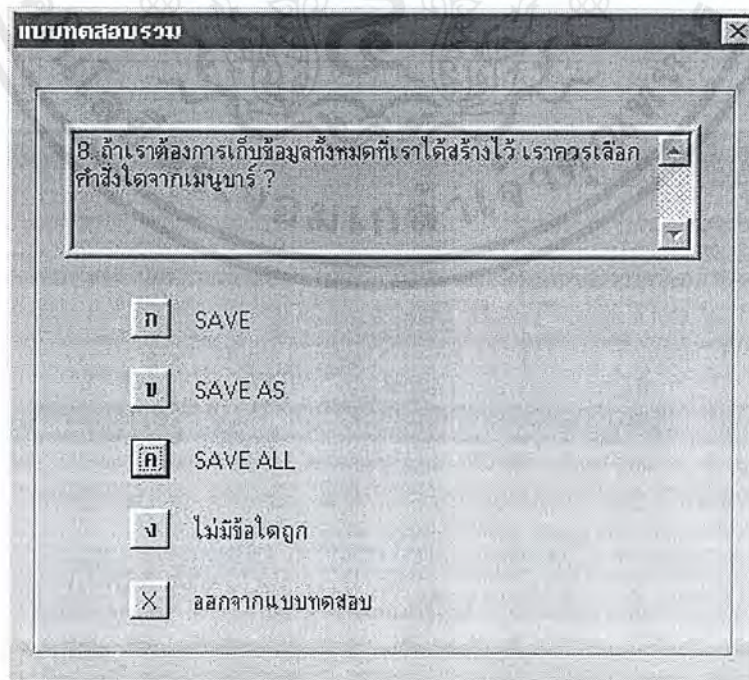
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อคลิกปุ่ม “Yes” ก็จะพบไดอะล็อกแสดงรายละเอียดในการทำแบบทดสอบดังรูปที่ 4.27 แต่ถ้าวคลิกปุ่ม “No” ก็จะกลับสู่เมนูหลัก



รูปที่ 4.27 รายละเอียดในการทำข้อสอบ

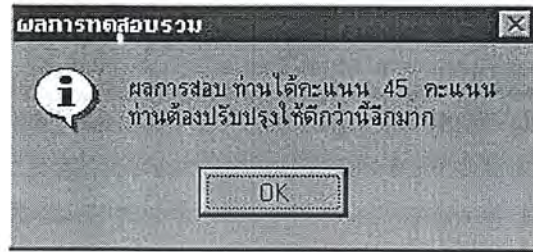
เมื่อคลิก OK จะพบกับแบบทดสอบดังรูปที่ 4.28



รูปที่ 4.28 แบบทดสอบรวม

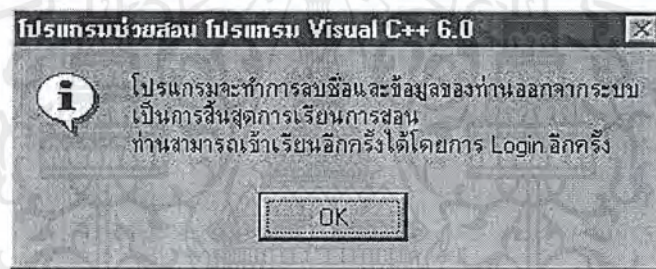
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานในท้องถิ่นเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้เรียนทำข้อสอบเสร็จแล้วจะมีการประเมินผลในการทำข้อสอบ ดังรูปที่ 4.29



รูปที่ 4.29 ผลการทำแบบทดสอบรวม

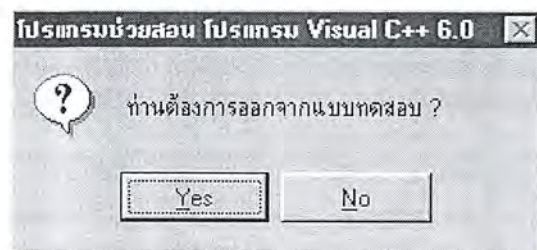
เมื่อคลิก OK แล้วจะพบกับไดอะล็อกดังรูปที่ 4.30



รูปที่ 4.30 ไดอะล็อกเตือนผู้เรียน

เมื่อสิ้นสุดการทำแบบทดสอบรวม โปรแกรมจะทำการลบข้อมูลทุกอย่างของผู้เรียนที่ทำแบบทดสอบออกจากระบบ เป็นการสิ้นสุดการเรียน

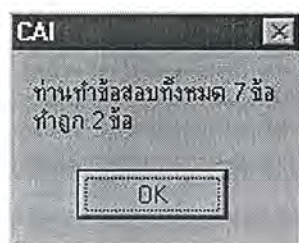
ถ้าต้องการออกจากโปรแกรมก่อนที่จะทำข้อสอบจนครบก็สามารถทำได้โดยการคลิกที่ปุ่มออกจากแบบทดสอบ และจะพบกับไดอะล็อกดังรูปที่ 4.31



รูปที่ 4.31 ไดอะล็อกถามความต้องการในการออกจากแบบทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อคลิกที่ปุ่ม Yes แล้วจะพบกับไดอะล็อกดังรูปที่ 4.32



รูปที่ 4.32 ไดอะล็อกแสดงผลการทำข้อสอบ

4.10 ผลการตอบแบบสอบถาม

จากการทดสอบโปรแกรมได้นำโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0 นี้ ไปทดลองใช้กับกลุ่มตัวอย่างที่เป็นนักศึกษาในระดับต่าง ๆ จำนวน 50 คน ได้ผลดังต่อไปนี้

4.10.1 ตอนที่ 1 ข้อมูลเกี่ยวกับสถานภาพของผู้ตอบ

ตารางที่ 4.1 สถานภาพของผู้ตอบ

สถานภาพ	จำนวน (คน)	จำนวน (%)
1. อาจารย์	0	0
2. นักศึกษา	50	100
รวม	50	100

ตารางที่ 4.2 เพศของผู้ตอบ

เพศ	จำนวน (คน)	จำนวน (%)
1. ชาย	27	54
2. หญิง	23	46
รวม	50	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 สถานศึกษาของผู้ตอบ

สถานศึกษา	จำนวน (คน)	จำนวน (%)
1. สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	27	54
2. มหาวิทยาลัยรามคำแหง	7	14
3. มหาวิทยาลัยเซนซ้อหิน	1	2
4. มหาวิทยาลัยเทคโนโลยีมนคร	3	6
5. สถาบันเทคโนโลยีราชมงคล วิทยาเขตพระนครเหนือ	1	2
6. สถาบันเทคโนโลยีราชมงคล วิทยาเขตภาคใต้สงขลา	1	2
7. สถาบันเทคโนโลยีราชมงคล วิทยาเขตวังไกลกังวล	1	2
8. วิทยาลัยเทคนิคสมุทรสงคราม	1	2
9. วิทยาลัยเทคนิคปทุมธานี	1	2
10. วิทยาลัยเทคนิคคอนเมือง	1	2
11. วิทยาลัยเทคนิคจะเชิงเตตรา	1	2
12. วิทยาลัยเทคนิคสัดหีบ	3	6
13. วิทยาลัยเทคนิคน่าน	1	2
14. วิทยาลัยเทคนิคหาดใหญ่	1	2
รวม	50	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 แผนกวิชาของผู้ตอบ

แผนก	จำนวน (คน)	จำนวน (%)
1. อิเล็กทรอนิกส์และคอมพิวเตอร์	24	48
2. วิศวกรรมโทรคมนาคม	4	8
3. เทคโนโลยีการวัดคุมทางอุตสาหกรรม	1	2
4. เทคโนโลยีการผลิตสัตว์	1	2
5. บริหารธุรกิจ	4	8
6. รัฐศาสตร์	1	2
7. มนุษย์ศาสตร์	2	4
8. วิทยาศาสตร์	1	2
9. อิเล็กทรอนิกส์	7	14
10. เทคนิคคอมพิวเตอร์	3	6
11. การโรงแรม	1	2
12. ไฟฟ้ากำลัง	1	2
รวม	50	100

ตารางที่ 4.5 ระดับการศึกษาของผู้ตอบ

ระดับการศึกษา	จำนวน (คน)	จำนวน (%)
1. ปริญญาตรี	39	78
2. ปวส.	10	20
3. ปวช.	1	2
รวม	50	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.6 ประสิทธิภาพในการใช้สื่อการสอนของผู้ตอบ

ประสิทธิภาพในการใช้สื่อการสอน	จำนวน (คน)	จำนวน (%)
1. เคยใช้	12	24
2. ไม่เคยใช้	38	76
รวม	50	100

4.10.2 ตอนที่ 2 ข้อมูลเกี่ยวกับประสิทธิภาพของโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0

ตารางที่ 4.7 ประสิทธิภาพการใช้สื่อการสอนด้านทักษะความรู้

ประสิทธิภาพการใช้สื่อ การสอนโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0	ระดับคะแนน											
	5		4		3		2		1		รวม	
	คน	%	คน	%	คน	%	คน	%	คน	%	คน	%
1. ด้านทักษะ ความรู้												
1.1 เนื้อหารายวิชาครบถ้วน สอดคล้องตามหลักสูตร	14	28	24	48	12	24	0	0	0	0	50	100
1.2 การบอกรับทฤษฎี ของเนื้อหา	9	18	31	62	9	18	1	2	0	0	50	100
1.3 การนำเข้าสู่บทเรียน	7	14	28	56	15	30	0	0	0	0	50	100
1.4 เนื้อหาที่สอนเป็นการ เรียนรู้ทางทฤษฎี	10	20	29	58	11	22	0	0	0	0	50	100
1.5 เนื้อหาที่สอนเป็นการ ฝึกทักษะ	9	18	28	56	11	22	2	4	0	0	50	100
1.6 การเรียงลำดับเนื้อหา รายวิชา	14	28	23	46	13	26	0	0	0	0	50	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.7 (ต่อ) ประสิทธิภาพการใช้สื่อการสอนด้านทักษะความรู้

ประสิทธิภาพการใช้สื่อ การสอนโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0	ระดับคะแนน											
	5		4		3		2		1		รวม	
	คน	%	คน	%	คน	%	คน	%	คน	%	คน	%
1.7 การเรียงลำดับเนื้อหาใน แต่ละบทเรียน	11	22	27	54	12	24	0	0	0	0	50	100
1.8 การเน้นส่วนสำคัญของ เนื้อหารายวิชา	12	24	19	38	15	30	4	8	0	0	50	100
1.9 การเรียนรู้จากการอ่านคำ อธิบายบนจอภาพ	13	26	17	34	18	36	1	2	1	2	50	100
1.10 การเรียนรู้จากการฟังคำ บรรยายของโปรแกรม	15	30	20	40	14	28	1	2	0	0	50	100
1.11 การเรียนรู้จากภาพเคลื่อนไหว ของโปรแกรม	9	18	16	32	21	42	4	8	0	0	50	100
1.12 ตัวอย่างประกอบคำ อธิบายเนื้อหาวิชา	10	20	21	42	18	36	1	2	0	0	50	100
1.13 การสรุปบทวนเนื้อหา ในแต่ละบทเรียน	9	18	23	46	15	30	3	6	0	0	50	100
1.14 การตั้งคำถามในแบบ ทดสอบ	13	26	20	40	14	28	3	6	0	0	50	100
รวม	155	22.1	326	46.6	198	28.3	20	2.86	1	0.14	700	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.8 ประสิทธิภาพการใช้สื่อการสอนด้านการใช้งาน

ประสิทธิภาพการใช้สื่อ การสอน โปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0	ระดับคะแนน											
	5		4		3		2		1		รวม	
	คน	%	คน	%	คน	%	คน	%	คน	%	คน	%
2. ด้านการใช้งาน												
2.1 เป็นการสื่อสารแบบสอง ทาง	16	32	22	44	11	22	1	2	0	0	50	100
2.2 สร้างความเข้าใจในการ ศึกษาบทเรียน	7	14	22	44	16	32	4	8	1	2	50	100
2.3 ฟังก์ชันอำนวยความสะดวก ในการใช้งาน โปรแกรม	7	14	21	42	19	38	0	0	3	6	50	100
2.4 การโต้ตอบระหว่างผู้ เรียนกับโปรแกรมขณะ ใช้งาน	12	24	16	32	18	36	4	8	0	0	50	100
2.5 ขนาดของตัวอักษรบนจอ ภาพ	18	36	21	42	10	20	1	2	0	0	50	100
2.6 รูปแบบและสีสันของตัว อักษร	14	28	17	34	14	28	1	2	4	8	50	100
2.7 ความชัดเจนของเสียง บรรยาย	17	34	18	36	13	26	2	4	0	0	50	100
2.8 ความถูกต้องของอักขระ คำบรรยาย	14	28	19	38	16	32	1	2	0	0	50	100
2.9 การเน้นคำบรรยาย	15	30	20	40	14	28	1	2	0	0	50	100
2.10 การเว้นช่วงการพูด	9	18	19	38	21	42	1	2	0	0	50	100
2.11 การใช้คำพูดในการ บรรยาย	8	16	21	42	19	38	1	2	1	2	50	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.8 (ต่อ) ประสิทธิภาพการใช้สื่อการสอนด้านการใช้งาน

ประสิทธิภาพการใช้สื่อ การสอนโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0	ระดับคะแนน											
	5		4		3		2		1		รวม	
	คน	%	คน	%	คน	%	คน	%	คน	%	คน	%
2.12 การอธิบายโดยใช้ภาพ เคลื่อนไหว	9	18	19	38	17	34	5	10	0	0	50	100
2.13 การใช้สีสังเกตุกับจอ ภาพ	9	18	14	28	17	34	5	10	5	10	50	100
2.14 การเสริมกำลังใจในการ เรียน	5	10	24	48	17	34	3	6	1	2	50	100
2.15 เครื่องมืออำนวยความสะดวก ในการทำแบบ ทดสอบ	7	14	19	38	22	44	2	4	0	0	50	100
2.16 การวัดผลและรายงานผล การทำแบบทดสอบ	6	12	22	44	21	42	1	2	0	0	50	100
รวม	173	21.6	314	39.3	265	33.1	33	4.13	15	1.88	800	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.9 ประสิทธิภาพการใช้สื่อการสอนด้านประสิทธิภาพอื่น ๆ

ประสิทธิภาพการใช้สื่อ การสอนโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0	ระดับคะแนน											
	5		4		3		2		1		รวม	
	คน	%	คน	%	คน	%	คน	%	คน	%	คน	%
3. ด้านประสิทธิภาพอื่นๆ												
3.1 เหมาะสมกับการศึกษา ด้วยตนเอง	19	38	20	40	10	20	0	0	1	2	50	100
3.2 ไม่จำกัดสถานที่ที่จะทำ การศึกษา	19	38	16	32	12	24	2	4	1	2	50	100
3.3 ทำให้ผู้เรียน เกิดการ เรียนรู้ได้เร็วขึ้น	19	38	18	36	12	24	0	0	1	2	50	100
3.4 ทำให้ผู้เรียน จำบทเรียน ได้นานขึ้น	13	26	20	40	15	30	2	4	0	0	50	100
3.5 ประหยัดเวลาในการ ศึกษาบทเรียน	17	34	22	44	9	18	2	4	0	0	50	100
3.6 เกิดความคิดสร้างสรรค์	13	26	21	42	14	28	2	4	0	0	50	100
3.7 การใช้งานบนเครื่อง คอมพิวเตอร์รุ่นปัจจุบัน	18	36	20	40	11	22	1	2	0	0	50	100
3.8 ความง่ายในการใช้งาน	15	30	23	46	12	24	0	0	0	0	50	100
3.9 ความเหมาะสมกับการ พัฒนาสู่ระบบการเรียน การสอน	14	28	22	44	12	24	2	4	0	0	50	100
รวม	147	32.7	182	40.4	107	23.8	11	2.44	3	0.67	450	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.10 ประสิทธิภาพการใช้สื่อการสอนการหาค่าเฉลี่ยด้านทักษะความรู้

ประสิทธิภาพการใช้สื่อ การสอนโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0	ผลการวิจัย		
	\bar{x}	ระดับคะแนน	อันดับที่
1. ด้านทักษะ ความรู้			
1.1 เนื้อหารายวิชาครบถ้วน สอดคล้องตามหลักสูตร	3.8	ดี	1
1.2 การบอกวัตถุประสงค์ ของเนื้อหา	3.76	ดี	2
1.3 การนำเข้าสู่บทเรียน	3.54	ดี	6
1.4 เนื้อหาที่สอนเป็นการ เรียนรู้ทางทฤษฎี	3.76	ดี	2
1.5 เนื้อหาที่สอนเป็นการ ฝึกทักษะ	3.62	ดี	5
1.6 การเรียงลำดับเนื้อหา รายวิชา	3.76	ดี	2
1.7 การเรียงลำดับเนื้อหาใน แต่ละบทเรียน	3.74	ดี	3
1.8 การเน้นส่วนสำคัญของ เนื้อหารายวิชา	3.4	พอใช้	9
1.9 การเรียนรู้จากการอ่านคำ อธิบายบนจอภาพ	3.4	พอใช้	9
1.10 การเรียนรู้จากการฟังคำ บรรยายของโปรแกรม	3.68	ดี	4
1.11 การเรียนรู้จากภาพเคลื่อนไหว ของโปรแกรม	3.1	พอใช้	10
1.12 ตัวอย่างประกอบคำ อธิบายเนื้อหารายวิชา	3.42	พอใช้	8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.10 (ต่อ) ประสิทธิภาพการใช้สื่อการสอนการหาค่าเฉลี่ยด้านทักษะความรู้

ประสิทธิภาพการใช้สื่อ การสอนโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0	ผลการวิจัย		
	\bar{x}	ระดับคะแนน	อันดับที่
1.13 การสรุปบททวนเนื้อหา ในแต่ละบทเรียน	3.4	พอใช้	9
1.14 การตั้งคำถามในแบบ ทดสอบ	3.52	ดี	7
รวม	3.56	ดี	2

ตารางที่ 4.11 ประสิทธิภาพการใช้สื่อการสอนการหาค่าเฉลี่ยด้านการใช้งาน

ประสิทธิภาพการใช้สื่อ การสอนโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0	ผลการวิจัย		
	\bar{x}	ระดับคะแนน	อันดับที่
2. ด้านการใช้งาน			
2.1 เป็นการสื่อสารแบบสอง ทาง	3.82	ดี	2
2.2 สร้างความเข้าใจในการ ศึกษายบทเรียน	3.18	พอใช้	11
2.3 ฟังก์ชันอำนวยความสะดวก ในการใช้งาน โปรแกรม	3.14	พอใช้	13
2.4 การโต้ตอบระหว่างผู้ เรียนกับโปรแกรมขณะ ใช้งาน	3.28	พอใช้	7
2.5 ขนาดของตัวอักษรบนจอ	3.9	ดี	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.11 (ต่อ) ประสิทธิภาพการใช้สื่อการสอนการหาค่าเฉลี่ยด้านการใช้งาน

ประสิทธิภาพการใช้สื่อ การสอนโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0	ผลการวิจัย		
	\bar{x}	ระดับคะแนน	อันดับที่
2.6 รูปแบบและสีสันทันของตัวอักษร	3.34	พอใช้	6
2.7 ความชัดเจนของเสียงบรรยาย	3.7	ดี	3
2.8 ความถูกต้องของอักขระคำบรรยาย	3.58	ดี	5
2.9 การเน้นคำบรรยาย	3.68	ดี	4
2.10 การเว้นช่วงการพูด	3.28	พอใช้	7
2.11 การใช้คำพูดในการบรรยาย	3.26	พอใช้	8
2.12 การอธิบายโดยใช้ภาพเคลื่อนไหว	3.2	พอใช้	10
2.13 การใช้สีสันทันกราฟิกบนจอภาพ	2.8	พอใช้	14
2.14 การเสริมกำลังใจในการเรียน	3.16	พอใช้	12
2.15 เครื่องมืออำนวยความสะดวกในการทำแบบทดสอบ	3.14	พอใช้	13
2.16 การวัดผลและรายงานผลการทำงานแบบทดสอบ	3.22	พอใช้	9
รวม	3.35	พอใช้	3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.12 ประสิทธิภาพการใช้สื่อการสอนการหาค่าเฉลี่ยด้านประสิทธิภาพอื่น ๆ

ประสิทธิภาพการใช้สื่อ การสอนโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0	ผลการวิจัย		
	\bar{x}	ระดับคะแนน	อันดับที่
3. ด้านประสิทธิภาพอื่นๆ			
3.1 เหมาะสมกับการศึกษา ด้วยตนเอง	3.9	ดี	1
3.2 ไม่จำกัดสถานที่ที่จะทำ การศึกษา	3.7	ดี	4
3.3 ทำให้ผู้เรียน เกิดการ เรียนรู้ได้เร็วขึ้น	3.82	ดี	3
3.4 ทำให้ผู้เรียน จำบทเรียน ได้นานขึ้น	3.54	ดี	7
3.5 ประหยัดเวลาในการ ศึกษาบทเรียน	3.86	ดี	2
3.6 เกิดความคิดสร้างสรรค์	3.58	ดี	6
3.7 การใช้งานบนเครื่อง คอมพิวเตอร์รุ่นปัจจุบัน	3.86	ดี	2
3.8 ความง่ายในการใช้งาน	3.82	ดี	3
3.9 ความเหมาะสมกับการ พัฒนาสู่ระบบการเรียน การสอน	3.68	ดี	5
รวม	3.75	ดี	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุป ปัญหา แนวทางแก้ไข และ พัฒนา

โปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0 นี้ ได้จัดทำขึ้นเพื่อให้ผู้เรียน ได้เรียนรู้เนื้อหาด้วยตนเองตามความสามารถ ความสนใจ และความพร้อมของผู้เรียนเอง ซึ่งเป็นจุดประสงค์หลักที่ต้องการจะนำไปใช้แทนการสอนแบบเดิม ซึ่งในเนื้อหาจะประกอบด้วย การติดตั้งโปรแกรม, ความรู้พื้นฐานเกี่ยวกับภาษา C++, อาร์เรย์, ฟอยเตอร์, การใช้แบบฟอร์มและเครื่องมือใน โปรแกรม Visual C++ 6.0 และการติดต่อกับไฟล์ข้อความ โดยโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0 นี้ จะทำขึ้นเพื่อให้ผู้เรียนได้เรียนรู้เนื้อหาด้วยตนเอง ตามความสามารถ ความสนใจและความพร้อมของผู้เรียน และยังมีแบบทดสอบเพื่อทดสอบความรู้ความเข้าใจในบทเรียนที่ได้ศึกษามาแล้วในแต่ละบท

ประสิทธิภาพในการทำงานของโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0 นี้ สามารถที่จะใช้งานได้ง่าย เพราะในโปรแกรมช่วยสอนจะมีส่วนที่แสดงให้เห็นทราบถึงข้อความ ที่จะเลือกศึกษาในแต่ละบทเรียนอย่างชัดเจน

แต่โครงการนี้ยังมีข้อบกพร่องและปัญหาที่เกิดขึ้นหลายประการ ทางด้านกลุ่มผู้จัดทำโครงการนี้ได้เขียนข้อเสนอแนะ เกี่ยวกับปัญหาที่เกิดขึ้นในช่วงเวลาที่ทำงาน และวิธีแก้ไขปัญหาที่เกิดขึ้น เพื่อที่จะเป็นประโยชน์ สำหรับผู้ที่จะนำโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0 นี้ ไปพัฒนาต่อดังรายละเอียด ต่อไปนี้

5.1 สรุปผลของโครงการ

โครงการโปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0 ที่ได้สร้างขึ้นนี้มีผลของโครงการนั้นสามารถบรรลุวัตถุประสงค์ที่ได้ตั้งไว้กล่าวคือ

1. สามารถนำโปรแกรมมาใช้ในการเรียนการสอนได้
2. สามารถออกแบบโปรแกรมช่วยสอนได้
3. สามารถสร้างโปรแกรมช่วยสอนได้
4. สามารถนำโปรแกรมช่วยสอนนี้ไปใช้สอนจริงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ปัญหาที่พบในการทำโครงการ

1. เครื่องคอมพิวเตอร์ของกลุ่มผู้จัดทำ มีความเร็วในการทำงานต่ำ โดยในช่วงขณะที่ทำการรันโปรแกรม ScreenCam พร้อมกับ โปรแกรม Visual C++ 6.0 จะทำให้เครื่องคอมพิวเตอร์มีการทำงานที่ช้ามาก
2. พื้นที่ความจุในฮาร์ดิสก์ของผู้จัดทำ มีขนาดเล็กไม่เพียงพอเก็บข้อมูลที่ทำในโปรแกรม เนื่องจากข้อมูลของโปรแกรมมีขนาดใหญ่มาก เพราะมีข้อมูลทั้งภาพและเสียง
3. ข้อมูลที่อยู่ในฮาร์ดิสก์เกิดการสูญหายพร้อมกับข้อมูลอื่นๆ สาเหตุที่เกิดขึ้นคาดว่าเกิดจากไวรัสคอมพิวเตอร์
4. การทำเนื้อหาแต่ละเรื่องต้องทำความเข้าใจในเนื้อหาซึ่งเสียเวลาในการศึกษาเนื้อหา
5. ฟังก์ชันบางฟังก์ชันของภาษา C ไม่สามารถนำมาใช้กับโปรแกรม Visual C++ 6.0 ได้ทั้งหมดเพราะจะทำให้เกิดการผิดพลาดขณะทำการคอมไพล์ได้

5.3 การแก้ปัญหา

1. เครื่องคอมพิวเตอร์มีความเร็วต่ำนั้น ได้แก้ปัญหาโดยการเพิ่มหน่วยความจำ (RAM) ขึ้นอีก 64 Mbytes รวมเป็น 96 Mbytes
2. พื้นที่ความจุในฮาร์ดิสก์ไม่เพียงพอ จึงทำการจัดซื้อใหม่อีก 3 GB
3. ทำการติดตั้งโปรแกรมป้องกันไวรัสรุ่นล่าสุดแทนโปรแกรมเดิม
4. ในการทำความเข้าใจเนื้อหาให้เร็วขึ้นนั้นทำได้โดยทำการอ่านก่อนให้พอเข้าใจก่อนแล้วจึงไปปรึกษากับผู้รู้
5. ทำการปรับปรุงโปรแกรมใหม่โดยนำฟังก์ชันของ โปรแกรม Visual C++ 6.0 มาใช้แทน

5.4 แนวทางในการพัฒนา

1. พัฒนาโปรแกรมในส่วนเนื้อหาของบทเรียนให้มากยิ่งขึ้น และให้มีบทเรียนที่มีความเหมาะสมมากขึ้น
2. เพิ่มส่วนของเสียงประกอบกับบทเรียนเพื่อให้บทเรียนมีความน่าสนใจมากยิ่งขึ้น
3. เพิ่มแบบทดสอบในบทเรียนมากขึ้น เพื่อให้ให้นักเรียนได้มีความรู้ความเข้าใจในเนื้อหาและบทเรียนมากขึ้น
4. ปรับปรุงแบบของโปรแกรมให้มีสีสันน่าใช้มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 รายงานการวิจัย

ในการทำโครงการครั้งนี้ ผู้จัดทำได้จัดทำแบบสอบถามเพื่อทำการวิจัยประสิทธิภาพของโครงการ โดยได้นำแบบสอบถามไปทำการสอบถามกับกลุ่มตัวอย่างจำนวน 50 คน

เครื่องมือที่ใช้ในการทำวิจัย เป็นแบบสอบถามแบ่งออกเป็น 2 ตอนคือ ข้อมูลเกี่ยวกับสถานะภาพของผู้ตอบ และข้อมูลเกี่ยวกับประสิทธิภาพการใช้สื่อการสอน

5.6 ผลการวิจัยแบบสอบถาม

5.6.1 ตอนที่ 1 ข้อมูลเกี่ยวกับสถานะภาพของผู้ตอบ

1. สถานะภาพทั้งหมดเป็นนักศึกษา
2. เพศ โดยรวมจะเป็นเพศชายมากกว่าเพศหญิง
3. สถานศึกษา ส่วนใหญ่จะเป็นนักศึกษาสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
4. แผนกวิชา นักศึกษาส่วนใหญ่จะเป็นนักศึกษาสาขาอิเล็กทรอนิกส์และคอมพิวเตอร์
5. ระดับการศึกษา ส่วนใหญ่เป็นนักศึกษาระดับปริญญาตรี
6. ประสบการณ์ในการใช้โปรแกรมช่วยสอน ส่วนใหญ่จะเคยใช้

5.6.2 ตอนที่ 2 ข้อมูลเกี่ยวกับประสิทธิภาพการใช้สื่อการสอน

- 1) ผลการวิจัยด้านทักษะความรู้ มีระดับความคิดเห็นโดยรวมอยู่ในระดับดี ($\bar{x} = 3.56$) ซึ่งมีรายละเอียดการจัดอันดับ จำแนกรายข้อ ดังนี้

อันดับที่ 1 ($\bar{x} = 3.8$) เนื้อหารายวิชาครบถ้วนสอดคล้องตามหลักสูตร จัดอยู่ในระดับดี

อันดับที่ 2 ($\bar{x} = 3.76$) เนื้อหาที่สอนเป็นการเรียนรู้ทางทฤษฎี และการเรียงลำดับเนื้อหา

รายวิชา จัดอยู่ในระดับดี

อันดับที่ 3 ($\bar{x} = 3.74$) การเรียงลำดับเนื้อหาในแต่ละบทเรียน จัดอยู่ในระดับดี

อันดับที่ 4 ($\bar{x} = 3.68$) การเรียนรู้จากการฟังคำบรรยายของโปรแกรม จัดอยู่ในระดับดี

อันดับที่ 5 ($\bar{x} = 3.62$) เนื้อหาที่สอนเป็นการฝึกทักษะ จัดอยู่ในระดับดี

อันดับที่ 6 ($\bar{x} = 3.54$) การนำเข้าสู่บทเรียน จัดอยู่ในระดับดี

อันดับที่ 7 ($\bar{x} = 3.52$) การตั้งคำถามในแบบทดสอบ จัดอยู่ในระดับดี

อันดับที่ 8 ($\bar{x} = 3.42$) ตัวอย่างประกอบคำอธิบายเนื้อหาวิชา จัดอยู่ในระดับพอใช้

อันดับที่ 9 ($\bar{x} = 3.4$) การเน้นส่วนสำคัญของเนื้อหาวิชา, การเรียนรู้จากการอ่านคำ

อธิบายบนจอภาพ และการสรุปบทวนเนื้อหาในแต่ละบทเรียน จัดอยู่ในระดับพอใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อันดับที่ 10 ($\bar{x} = 3.1$) การเรียนรู้จากภาพเคลื่อนไหวของโปรแกรม จัดอยู่ในระดับพอใช้

2) ผลการวิจัยด้านการใช้งาน มีระดับความคิดเห็นโดยรวมอยู่ในระดับพอใช้ ($\bar{x} = 3.35$) ซึ่งมีรายละเอียดการจัดอันดับ จำแนกรายข้อ ดังนี้

อันดับที่ 1 ($\bar{x} = 3.9$) ขนาดของตัวอักษรบนจอภาพ จัดอยู่ในระดับดี

อันดับที่ 2 ($\bar{x} = 3.82$) เป็นการสื่อสารแบบสองทาง จัดอยู่ในระดับดี

อันดับที่ 3 ($\bar{x} = 3.7$) ความชัดเจนของเสียงบรรยาย จัดอยู่ในระดับดี

อันดับที่ 4 ($\bar{x} = 3.68$) การเน้นคำบรรยาย จัดอยู่ในระดับดี

อันดับที่ 5 ($\bar{x} = 3.58$) ความถูกต้องของอักขระคำบรรยาย จัดอยู่ในระดับดี

อันดับที่ 6 ($\bar{x} = 3.34$) รูปแบบและสีสันทของตัวอักษร จัดอยู่ในระดับพอใช้

อันดับที่ 7 ($\bar{x} = 3.28$) การโต้ตอบระหว่างผู้เรียนกับโปรแกรมขณะใช้งาน และ

การเว้นช่วงการพูด จัดอยู่ในระดับพอใช้

อันดับที่ 8 ($\bar{x} = 3.26$) การใช้คำพูดในการบรรยาย จัดอยู่ในระดับพอใช้

อันดับที่ 9 ($\bar{x} = 3.22$) การวัดผลและรายงานผลการทำแบบทดสอบ จัดอยู่ในระดับพอใช้

อันดับที่ 10 ($\bar{x} = 3.2$) การอธิบายโดยใช้ภาพเคลื่อนไหว จัดอยู่ในระดับพอใช้

อันดับที่ 11 ($\bar{x} = 3.18$) สร้างความเข้าใจในการศึกษาบทเรียน จัดอยู่ในระดับพอ

ใช้

อันดับที่ 12 ($\bar{x} = 3.16$) การเสริมกำลังใจในการเรียน จัดอยู่ในระดับพอใช้

อันดับที่ 13 ($\bar{x} = 3.14$) ฟังก์ชันอำนวยความสะดวกในการใช้งาน โปรแกรม และ เครื่องมืออำนวยความสะดวกในการทำแบบทดสอบ จัดอยู่ในระดับพอใช้

อันดับที่ 14 ($\bar{x} = 2.8$) การใช้สีสันทกราฟิกบนจอภาพ จัดอยู่ในระดับพอใช้

3) ผลการวิจัยด้านประสิทธิภาพอื่น ๆ มีระดับความคิดเห็นโดยรวมอยู่ในระดับดี

($\bar{x} = 3.75$) ซึ่งมีรายละเอียดการจัดอันดับ จำแนกรายข้อ ดังนี้

อันดับที่ 1 ($\bar{x} = 3.9$) เหมาะสมกับการศึกษด้วยตนเอง จัดอยู่ในระดับดี

อันดับที่ 2 ($\bar{x} = 3.86$) ประหยัดเวลาในการศึกษาบทเรียน และการทำงานบน เครื่องคอมพิวเตอร์รุ่นปัจจุบัน จัดอยู่ในระดับดี

อันดับที่ 3 ($\bar{x} = 3.82$) ทำให้ผู้เรียน เกิดการเรียนรู้ได้เร็วขึ้น และความง่ายในการ ใช้งาน จัดอยู่ในระดับดี

อันดับที่ 4 ($\bar{x} = 3.7$) ไม่จำกัดสถานที่ที่จะทำการศึกษา จัดอยู่ในระดับดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อันดับที่ 5 ($\bar{x} = 3.68$) ความเหมาะสมกับการพัฒนาสู่ระบบการเรียนการสอน จัดอยู่ในระดับดี

อันดับที่ 6 ($\bar{x} = 3.58$) เกิดความคิดสร้างสรรค์ จัดอยู่ในระดับดี

อันดับที่ 7 ($\bar{x} = 3.54$) ทำให้ผู้เรียน จำบทเรียน ได้นานขึ้น จัดอยู่ในระดับดี

5.7 อภิปรายผลการวิจัย

การอภิปรายผลการวิจัยต่อไปนี้เป็น การอภิปรายผลจากภาพรวม โดยการรวบรวมข้อมูลของการวิจัยที่น่าสนใจเป็นหัวข้อดังต่อไปนี้

5.7.1 ข้อมูลเกี่ยวกับสถานภาพของผู้ตอบ

กลุ่มประชากรตัวอย่าง ส่วนใหญ่จะเป็นนักศึกษาเพศชาย สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ ระดับปริญญาตรี สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งส่วนใหญ่จะมีประสบการณ์ในการใช้สื่อการสอนประเภทโปรแกรมช่วยสอนมาแล้ว

5.7.2 ข้อมูลเกี่ยวกับประสิทธิภาพการใช้สื่อการสอน

- 1) ด้านทักษะความรู้ โดยรวมจัดอยู่ในระดับดี หัวข้อเนื้อหาวิชาครบถ้วนสอดคล้องตามหลักสูตรได้รับความคิดเห็นมากที่สุด รองลงมาเป็นหัวข้อเนื้อหาที่สอนเป็นการเรียนรู้ทางทฤษฎี และหัวข้อการเรียงลำดับเนื้อหาวิชา
- 2) ด้านการใช้งาน โดยรวมจัดอยู่ในระดับพอใช้ ขนาดของตัวอักษรบนจอภาพได้รับความคิดเห็นมากที่สุด รองลงมาเป็นการสื่อสารแบบสองทาง
- 3) ด้านประสิทธิภาพอื่น ๆ โดยรวมจัดอยู่ในระดับดี เหมาะสมกับการศึกษาด้วยตนเองได้รับความคิดเห็นมากที่สุด รองลงมาเป็นประหยัดเวลาในการศึกษาบทเรียน และการใช้งานบนเครื่องคอมพิวเตอร์รุ่นปัจจุบัน



ภาคผนวก ก
แบบสอบถามประสิทธิภาพการใช้สื่อการสอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบสอบถามประสิทธิภาพการใช้สื่อการสอน
โปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0
(Computer Assisted Instructional in Visual C++ 6.0)
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

แบบสอบถามนี้จัดทำขึ้น โดยมีวัตถุประสงค์เพื่อใช้ในการศึกษาประสิทธิภาพของการใช้สื่อการสอนด้วยเครื่องคอมพิวเตอร์ในการศึกษา ซึ่งเป็นส่วนหนึ่งของรายวิชา 03322210 โครงการสร้างอุปกรณ์เพื่อการสอน ในหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ทั้งนี้ผู้จัดทำโปรแกรมช่วยสอน โปรแกรม Visual C++ 6.0 ได้จัดทำแบบสอบถามขึ้นเพื่อใช้เป็นข้อมูลส่วนหนึ่งในการเขียนปฏิญานិพนธ์ประกอบกับโครงการฯ ซึ่งเป็นประโยชน์ต่อระบบการเรียนการสอนต่อไปในภายภาคหน้า ผู้จัดทำขอขอบพระคุณอย่างสูงในความกรุณาให้ความร่วมมือมา ณ โอกาสนี้ด้วย

(นาย ชัยสิทธิ แสงไทยเจริญลาภ)

(นางสาวอังคณา เสมาทอง)

ผู้จัดทำ

(อ.อมรชัย ชัยชนะ)

อาจารย์ที่ปรึกษาปฏิญานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอนที่ 1 ข้อมูลเกี่ยวกับสถานภาพของผู้ตอบ

คำชี้แจง โปรดเติมคำลงในช่องว่างหรือทำเครื่องหมาย ✓ ลงใน หน้าข้อความที่ตรงกับความเป็นจริง

1. สถานภาพ อาจารย์ นักศึกษา
2. เพศ ชาย หญิง
3. สถานศึกษา วิทยาลัยเทคนิค.....
 สถาบันเทคโนโลยีราชมงคล วิทยาเขต.....
 อื่นๆ (ระบุ)
4. แผนกวิชา อิเล็กทรอนิกส์ เทคนิคคอมพิวเตอร์
 ไฟฟ้ากำลัง อื่นๆ (ระบุ)
5. ระดับการศึกษา ปวช. ปวส.
 ปริญญาตรี อื่นๆ (ระบุ)
6. เคยมีประสบการณ์ในการใช้สื่อการสอนโปรแกรมช่วยสอนหรือไม่
 เคยใช้ ไม่เคยใช้

ตอนที่ 2 ข้อมูลเกี่ยวกับประสิทธิภาพการใช้สื่อการสอน

คำชี้แจง โปรดอ่านข้อความจากแบบสอบถามอย่างละเอียด และพิจารณาว่าท่านมีความคิดเห็นระดับใดหลังจากได้ทดลองใช้สื่อการสอน แล้วตอบแบบสอบถามความคิดเห็นหรือความรู้สึกรของท่าน โดยทำเครื่องหมาย ✓ ลงในช่องระดับคะแนนทางขวามือเพียงช่องเดียว

ความหมายของระดับคะแนน	ระดับคะแนน	5	หมายถึง	ดีมาก
	ระดับคะแนน	4	หมายถึง	ดี
	ระดับคะแนน	3	หมายถึง	ปานกลาง
	ระดับคะแนน	2	หมายถึง	พอใช้
	ระดับคะแนน	1	หมายถึง	ต้องปรับปรุง

ประสิทธิภาพการใช้สื่อการสอน โปรแกรมช่วยสอนโปรแกรม Visual C++ 6.0	ระดับคะแนน				
	5	4	3	2	1
1. ด้านทักษะ ความรู้					
1.1 เนื้อหารายวิชาครบถ้วนสอดคล้องตามหลักสูตร					
1.2 การบอกวัตถุประสงค์ของเนื้อหา					
1.3 การนำเข้าสู่บทเรียน					
1.4 เนื้อหาที่สอนเป็นการเรียนรู้ทางทฤษฎี					
1.5 เนื้อหาที่สอนเป็นการฝึกทักษะ					
1.6 การเรียงลำดับเนื้อหารายวิชา					
1.7 การเรียงลำดับเนื้อหาในแต่ละบทเรียน					
1.8 การเน้นส่วนสำคัญของเนื้อหารายวิชา					
1.9 การเรียนรู้จากการอ่านคำอธิบายบนจอภาพ					
1.10 การเรียนรู้จากการฟังคำบรรยายของโปรแกรม					
1.11 การเรียนรู้จากภาพเคลื่อนไหวของโปรแกรม					
1.12 ตัวอย่างประกอบคำอธิบายเนื้อหารายวิชา					
1.13 การสรุปบทวนเนื้อหาในแต่ละบทเรียน					
1.14 การตั้งคำถามในแบบทดสอบ					
2. ด้านการใช้งาน					
2.1 เป็นการสื่อสารแบบสองทาง					
2.2 สร้างความเข้าใจในการศึกษาบทเรียน					
2.3 ฟังก์ชันอำนวยความสะดวกในการใช้งาน โปรแกรม					
2.4 การโต้ตอบระหว่างผู้เรียนกับ โปรแกรมขณะใช้งาน					
2.5 ขนาดของตัวอักษรบนจอภาพ					
2.6 รูปแบบและสีสันของตัวอักษร					
2.7 ความชัดเจนของเสียงบรรยาย					
2.8 ความถูกต้องของอักขระคำบรรยาย					
2.9 การเน้นคำบรรยาย					
2.10 การเว้นช่วงการพูด					
2.11 การใช้คำพูดในการบรรยาย					
2.12 การอธิบายโดยใช้ภาพเคลื่อนไหว					
2.13 การใช้สีสันกราฟิกบนจอภาพ					
2.14 การเสริมกำลังใจในการเรียน					
2.15 เครื่องมืออำนวยความสะดวกในการทำแบบทดสอบ					
2.16 การวัดผลและรายงานผลการทำแบบทดสอบ					

เอกสารนี้เป็นเอกสารที่เผยแพร่ในนามของโรงเรียนการศึกษานานาชาติ ไม่อนุญาตให้นำไปใช้ ระบุชื่อด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- ชมธรรม อัญญาโชติ และสุภชัย เลิศผดุงสุข. “โปรแกรมช่วยสอนวิชาอิเล็กทรอนิกส์อุตสาหกรรม.”
 ปริญญาานิพนธ์ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์,
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2541
- นิรุช อำนวยศิลป์. คู่มือการเขียนโปรแกรม Microsoft Visual C++ Version 6.0. กรุงเทพฯ : บริษัท
 ชัคเซส มีเดีย จำกัด. 2542
- วิทยา เรื่องพรวิสุทธิ. คู่มือโปรแกรมภาษา C สำหรับผู้เริ่มต้น. กรุงเทพฯ : บริษัท เอช. เอ็น. กรุ๊ป
 จำกัด. 2539
- ดวงแก้ว สวามิภักดิ์. การโปรแกรมภาษา C. กรุงเทพฯ : บริษัท เอช. เอ็น. กรุ๊ป จำกัด. 2537
- ธันวา ศรีประโม่ง. การโปรแกรมภาษาซีสำหรับวิศวกรรม. กรุงเทพฯ : โครงการตำราวิชาการ
 วิทยาลัยมหานคร. 2536
- บริษัท สยามคอมพิวเตอร์. การเขียนโปรแกรมภาษา C++. กรุงเทพฯ : หจก.ไทยเจริญการพิมพ์.
 2538

ประวัติผู้แต่ง



ชื่อผู้ทำปฏิญานิพนธ์	นายชัยสิทธิ แสงไทยเจริญลาภ
วันเดือนปีเกิด	6 ธันวาคม 2520
สถานที่เกิด	จังหวัดประจวบคีรีขันธ์
ภูมิลำเนาเดิม	182/65 ซ.8 อ่างน้ำ ถนนชมสินธุ์ ตำบลหัวหิน อำเภอหัวหิน จังหวัดประจวบคีรีขันธ์ 77110
ที่อยู่ปัจจุบัน	397/1 หมู่ 1 ซอยจินดาภิเษก 10 แขวงลาดกระบัง เขตลาดกระบัง กรุงเทพมหานคร 10520
โทรศัพท์	(02) 3268456 ต่อห้อง 19 เก้า
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนมัธยมศึกษาสาธิตการวิทยา
มัธยมศึกษาตอนต้น	โรงเรียนหัวหิน
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยวิชาชีพวังไกลกังวล
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคสมุทรสงคราม
ปริญญาตรี	สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	กองทุนกู้ยืมเพื่อการศึกษา
คติพจน์	จงใช้เวลาในวันนี้ให้คุ้มค่าที่สุด เพราะเราอาจไม่มีวันพรุ่งนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้พิมพ์ได้ขออนุญาตเผยแพร่ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปฏิญานิพนธ์	นางสาวอังคณา เสมาทอง
วันเดือนปีเกิด	12 ตุลาคม 2519
สถานที่เกิด	กรุงเทพมหานคร
ภูมิลำเนาเดิม	212/458 หมู่ 5 ถนนวิภาวดี-รังสิต แขวงทุ่งสองห้อง เขตหลักสี่ กรุงเทพมหานคร 10210
ที่อยู่ปัจจุบัน	212/458 หมู่ 5 ถนนวิภาวดี-รังสิต แขวงทุ่งสองห้อง เขตหลักสี่ กรุงเทพมหานคร 10210
โทรศัพท์	9846519
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนรัชรินทร์วิทยา
มัธยมศึกษาตอนต้น	โรงเรียนรัชรินทร์วิทยา
ประกาศนียบัตรวิชาชีพ (ปวช.)	โรงเรียนเซนจอห์นโปลิเทคนิค
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	โรงเรียนเทคโนโลยีพระรามหก
ปริญญาตรี	สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คติพจน์	อย่ากลัวกับการเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้