

พื้นฐานการเขียนโปรแกรมด้วย Visual Basic 6

1 ความหมายและความเป็นมาของ Visual Basic

ภาษา Visual Basic เป็นภาษาคอมพิวเตอร์ (Programming Language) ที่พัฒนาโดยบริษัท ไมโครซอฟท์ซึ่งเป็นบริษัทที่สร้างระบบปฏิบัติการ Windows ที่ใช้กันอย่างแพร่หลายในปัจจุบัน โดยตัวภาษามีรากฐานมาจากภาษาเบสิก (Basic) ซึ่งย่อมาจาก Beginner's All Purpose Symbolic (ชุดคำสั่งหรือ ภาษาคอมพิวเตอร์ สำหรับผู้เริ่มต้น) ภาษาเบสิกมีจุดเด่นคือ ผู้ที่ไม่มีพื้นฐานเกี่ยวกับการเขียนโปรแกรมเลย ก็สามารถเรียนรู้และนำไปใช้งานได้โดยง่ายและรวดเร็ว เมื่อเทียบกับภาษาอื่นๆ เช่น ภาษาซี (C) ภาษาปาสคาล (Pascal) หรือแอสเซมบลี (Assembler)

ไมโครซอฟท์ได้พัฒนาโปรแกรมภาษา Basic นับตั้งแต่ภาษา MBasic (Microsoft Basic), BasicA (Basic Advanced) และ Quick Basic ซึ่งได้คิดตั้งมาพร้อมกับระบบปฏิบัติการ MS Dos ในที่สุดใช้ชื่อว่า QBasic แต่ละเวอร์ชันที่ออกมาได้มีการเพิ่มคำสั่งต่างๆ เข้าไปโดยตลอด ซึ่งโปรแกรมเหล่านี้ล้วนแต่ทำงานใน Text Mode คือเป็นตัวอักษรล้วนๆ จนกระทั่งมีระบบปฏิบัติการ Windows ทางไมโครซอฟท์ก็ได้ปรับปรุงภาษา Basic ออกมาใหม่เพื่อสนับสนุนการทำงานบน Windows ทำให้ Visual Basic ถือกำเนิดตั้งแต่นั้น

Visual Basic 1.0 เป็นเวอร์ชันแรกออกมาเมื่อปี 1991 โดยในช่วงแรกนั้นยังไม่มีความแตกต่างจาก QBasic มากนัก แต่จะเน้นเครื่องมือที่ใช้ในการเขียนโปรแกรมบนวินโดวส์ ซึ่งได้รับความนิยมเป็นอย่างดี ไมโครซอฟท์จึงได้พัฒนา Visual Basic ให้ดีขึ้นเรื่อยๆ จนในปัจจุบันเวอร์ชันล่าสุดคือ Visual Basic 6.0 ออกมาในช่วงปี 1998 ได้เพิ่มความสามารถ ในการเขียนโปรแกรมติดต่อกับเครือข่ายอินเทอร์เน็ต ด้านการเชื่อมต่อกับระบบฐานข้อมูล รวมทั้งเครื่องมือ และการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ให้สมบูรณ์และสะดวกยิ่งขึ้น

2 ข้อดีของการเขียนโปรแกรมด้วย Visual Basic

ข้อดีของการเขียนโปรแกรมด้วย Visual Basic คือง่ายต่อการเรียนรู้ และเหมาะสำหรับผู้เริ่มต้น ทั้งในเรื่องของไวยากรณ์ของภาษา และเครื่องมือในการใช้งาน

ภาษา Basic เป็นภาษาที่มีคนเรียนรู้ และมีการใช้งานมากที่สุดในประวัติศาสตร์ ของคอมพิวเตอร์

ภาษา Visual Basic มีการพัฒนาอย่างต่อเนื่อง ทั้งในด้านการปรับปรุงประสิทธิภาพของตัวภาษาและความเร็วในการประมวลผล และในด้านความสามารถใหม่ๆ เช่น ความสามารถการติดต่อกับระบบฐานข้อมูล การเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต เป็นต้น

ผู้พัฒนาสำคัญของ Visual Basic ก็คือบริษัทไมโครซอฟท์ เราจึงมั่นใจได้ว่า Visual Basic จะยังมีการพัฒนา ปรับปรุง และคงอยู่ได้อีกนาน

3 ลักษณะการเขียนโปรแกรมของ Visual Basic

การเขียนโปรแกรมของ Visual Basic จะอยู่ในลักษณะของ Event-Driven คือเป็นการเขียนโปรแกรมที่ตอบสนองต่อการควบคุมเหตุการณ์ต่างๆ มากมาย ที่เกิดจากการกระทำของผู้ใช้ เป็นการเขียนโปรแกรมที่ผูกไว้กับเหตุการณ์ ที่สามารถเกิดขึ้นกับออบเจ็กต์ หรือส่วนประกอบต่างๆ ที่อยู่บนหน้าจอ

4 ความต้องการของระบบในการติดตั้งโปรแกรม Visual Basic 6.0

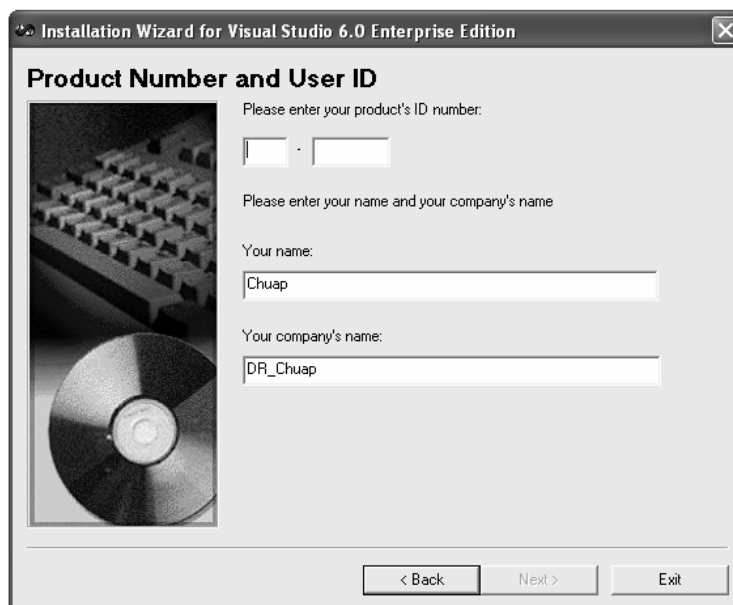
ในการใช้งาน Visual Basic 6.0 จะมีการต้องการระบบดังต่อไปนี้

- 1) ระบบปฏิบัติการ Windows 95/98 หรือ Windows NT หรือ Windows XP
- 2) เครื่องคอมพิวเตอร์ที่มี CPU. 486 DX/66 MHz เป็นอย่างต่ำขึ้นไป แต่ถ้าจะให้ทำงานได้ดี ควรจะเป็น CPU. รุ่น Pentium ขึ้นไป
- 3) ไดรฟ์ CD-ROM
- 4) จอภาพและการ์ดจอ ที่สนับสนุนการทำงานของ Windows

5 การติดตั้งโปรแกรม และการเรียกใช้งาน Visual Basic 6.0

การติดตั้งโปรแกรม Visual Basic 6.0 ก็เหมือนกับการติดตั้งโปรแกรมทั่วไปใน Windows คือโดยการเรียกไฟล์ Setup.exe หรือถ้าเป็นแบบ Auto Run ก็จะมีการเรียกไฟล์ Setup ให้โดยอัตโนมัติ ขั้นตอนการติดตั้ง Visual Basic 6.0 มีดังนี้

- 1) ใส่แผ่น Visual studio หรือแผ่น Visual Basic ที่ไดรฟ์ CD-ROM
- 2) รอให้เครื่องทำการ Auto Run แต่ถ้าโปรแกรมอัตโนมัติของเราไม่ทำงาน ให้ทำการเปิดไฟล์ Setup.exe
- 3) เมื่อมาถึงหน้าจอ Product Number and User ID ให้ใส่รหัสลงทะเบียน (CD-Key) ชื่อและบริษัทของคุณ



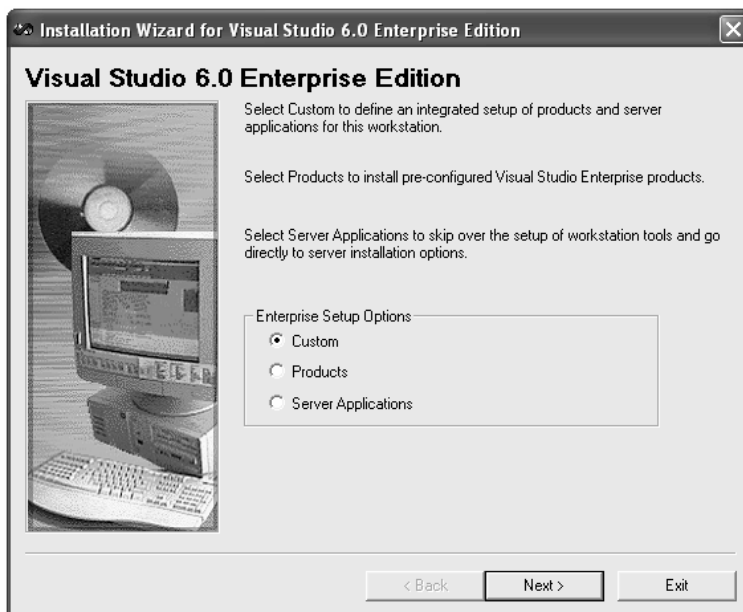
รูปที่ 2.1 หน้าต่าง Product Number and User ID

4) ในกรณีที่เครื่องของเรายังไม่มี Internet Explorer 4.0 โปรแกรมจะทำการติดตั้ง Internet Explorer ให้ก่อน โดยจะปรากฏหน้าจอ Install IE 4.01 ให้คลิกที่ช่อง **Install IE 4.01** แล้วคลิกที่ปุ่ม **Next**

5) ที่หน้าจอ Installation Option ในส่วนนี้โปรแกรมจะถามว่า ต้องการติดตั้งโปรแกรม Internet Explorer แบบ Standard หรือ Full ให้เราทำการเลือก แล้วคลิก **Next** ต่อไป

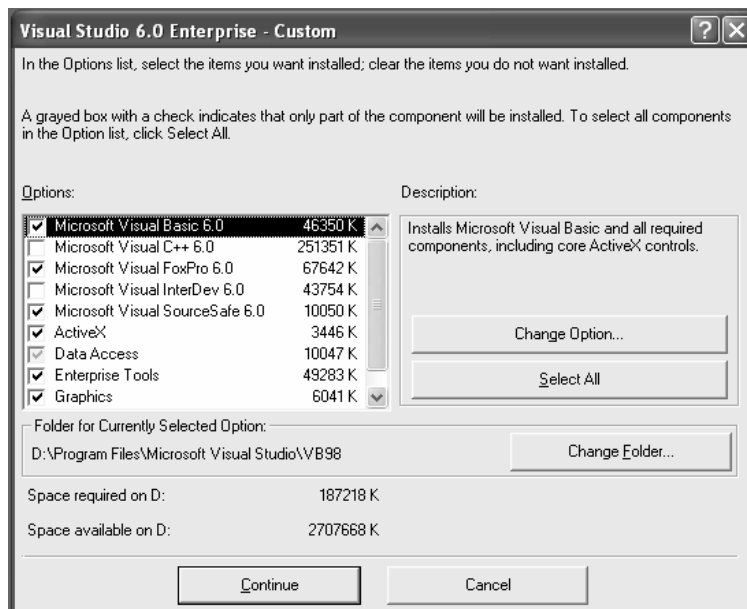
6) ที่หน้าจอ Windows Desktop Update เครื่องจะถามว่าต้องการเปลี่ยนเดสก์ท็อปของ Windows ให้ร่วมกับ Internet หรือไม่ ถ้าเครื่องที่มีหน่วยความจำน้อย ให้เลือกว่า “No” คือไม่ต้องการ Update desktop

7) เมื่อติดตั้ง Internet Explorer เสร็จแล้วต่อไปก็เริ่มเข้าสู่การติดตั้ง Visual Basic 6.0 ให้ทำการเลือกอปชัน (Option) **Custom** แล้วคลิกที่ปุ่ม **Next** ในส่วนหน้าของ Visual studio 6.0 Enterprise Edition



รูปที่ 2.2 หน้าต่าง Visual studio 6.0 Enterprise Edition

8) จากนั้นให้ทำการเลือกโปรแกรมที่ต้องการติดตั้ง



รูปที่ 2.3 หน้าต่าง เลือกโปรแกรมที่ต้องการติดตั้ง

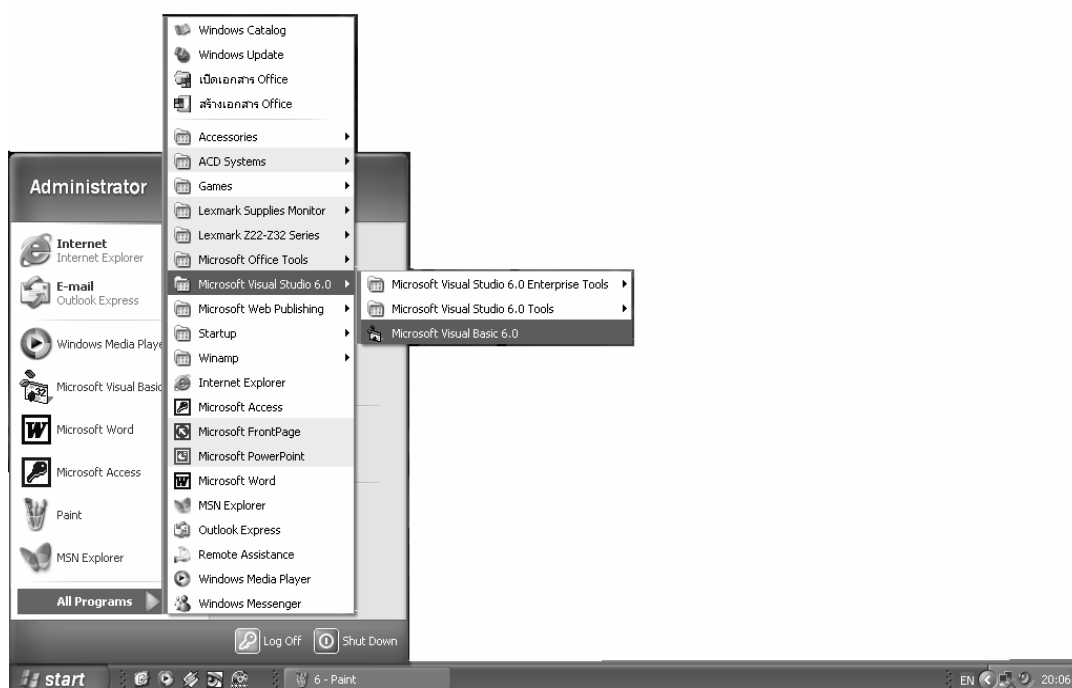
9) เมื่อติดตั้งเสร็จแล้ว จากนั้นเครื่องก็จะให้เราทำการ Restart Windows

10) เมื่อ Restart เครื่องและกลับสู่ Windows อีกครั้ง โปรแกรมก็จะถามว่าต้องการติดตั้ง Help หรือไม่ ถ้ามีแผ่น MSDN (Microsoft Developer Network Library Visual studio 6.0)

อยู่แล้ว ก็ให้เลือกตกลง แต่ถ้าไม่มีแผ่นก็ให้ออกจากการติดตั้ง MSDN ซึ่งเมื่อต้องการติดตั้ง Help ก็
สามารถติดตั้งจากแผ่น MSDN ได้โดยตรง

11) เมื่อทำการติดตั้งแล้วก็ยืนยันว่าเสร็จสิ้นขั้นตอนการติดตั้งโปรแกรม และพร้อมที่จะ
จะเข้าไปใช้งาน Visual Basic

ในการเรียกใช้โปรแกรม Visual Basic 6.0 สามารถทำได้โดยเลือกเมนู **Start>**
Program> Microsoft Visual Studio> Visual Basic 6.0



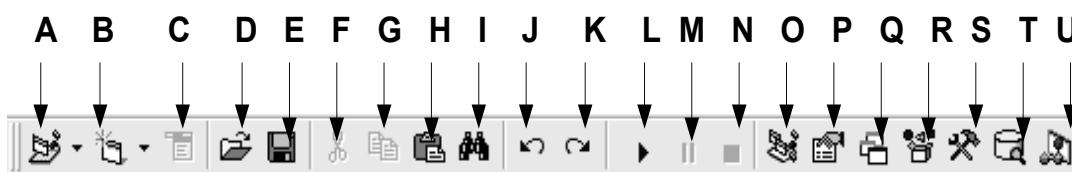
รูปที่ 2.4 หน้าต่าง การเรียกใช้งาน Visual Basic 6.0

6 ส่วนประกอบต่างๆ ที่ใช้ในการพัฒนาโปรแกรมของ Visual Basic 6 (IDE)

คำว่า IDE หรือ Integrated Development Environment คือ สภาพแวดล้อมการทำงานในการพัฒนาโปรแกรมโดยใช้ Visual Basic หรืออุปกรณ์เครื่องมือต่างๆที่เตรียมมาช่วยในการพัฒนาโปรแกรมด้วย Visual Basic เมื่อเปิดโปรแกรม Visual Basic ขึ้นมาครั้งแรก จะปรากฏหน้าจอของ IDE ซึ่งมีส่วนประกอบดังนี้

1) ทูลบาร์ (Tool Bar)

ทูลบาร์คือแผงควบคุม ที่มีปุ่มต่างๆ ประกอบอยู่ ซึ่งสามารถเรียกใช้ได้อย่างสะดวกและรวดเร็ว โดยรายละเอียดของปุ่มต่างๆ ของทูลบาร์มีดังนี้

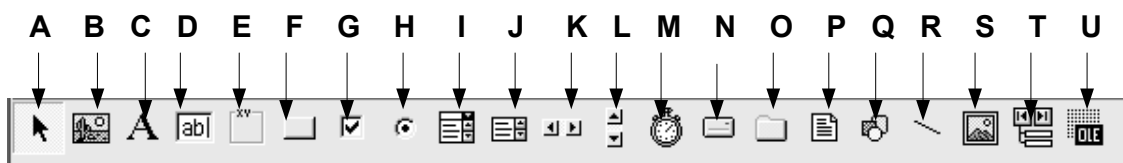


รูปที่ 2.5 เครื่องมือต่างๆ บน Tool Bar ของ Visual Basic 6.0

- A. ใช้สำหรับเปิดโปรเจ็คใหม่ขึ้นมา
- B. เพิ่มฟอร์ม โมดูล หรือออปเจ็ค ประเภทต่างๆ เขาไปใช้ในโปรเจ็ค หรือโปรแกรมที่เรา กำลังพัฒนาอยู่
- C. เปิด Menu Editor ซึ่งเป็นเครื่องมือที่ใช้ในการสร้างเมนูของ โปรแกรม
- D. เปิดไฟล์โปรเจ็ค
- E. บันทึกโปรเจ็ค Save
- F. ตัด(cut)
- G. ก๊อปปี้ (copy)
- H. วาง (paste)
- I. ค้นหา (Find)
- J. ยกเลิกการกระทำ (undo)
- K. เรียกคืนกับสิ่งที่ undo ไป (redo)
- L. สั่งให้โปรแกรมทำงาน (Run)
- M. ให้โปรแกรมหยุดทำงานชั่วคราว (Pause)
- N. ให้โปรแกรมหยุดทำงาน
- O. เปิดวินโดว์ Properties
- P. เปิดวินโดว์ From layout
- Q. Object Browser ช่วยในการค้นหารายละเอียดของออปเจ็คต่างๆ
- S. Toolbox เป็นแหล่งรวบรวมออปเจ็คต่างๆ
- T. เปิดวินโดว์ data view เพื่อดูการติดต่อกับฐานข้อมูลต่างๆ

2) ทุลบ็อก (Toolbox)

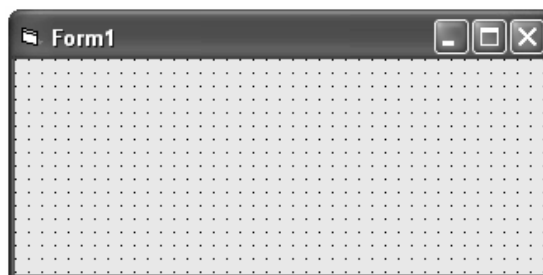
เป็นที่รวมออปเจ็คต่างๆ ที่จะนำมาใช้ในโปรแกรมซึ่งอาจเรียกว่า Control Object ซึ่งมีดัง ภาพต่อไปนี้ และนอกจากนี้เรายังสามารถเพิ่มออปเจ็คเข้าไปใน Toolbox ได้อีกมากมาย ซึ่ง รายละเอียดคร่าวๆ มีดังนี้



รูปที่ 2.6 เครื่องมือต่างๆ บน Toolbox ของ Visual Basic 6.0

- A. **Pointer** ใช้ในการจัดขนาด เคลื่อนย้าย และวางตำแหน่งของออปเจ็กต์ต่างๆ
- B. **Picture** ใช้ควบคุมและแสดงภาพต่างๆ ลงบนฟอร์ม
- C. **Label** ใช้แสดงข้อความต่างๆ ลงบนฟอร์ม
- D. **Textbox** ใช้สำหรับรับข้อความที่ผู้ใช้ป้อนเข้ามา
- E. **Frame** ใช้สำหรับจัดกลุ่มและรวมออปเจ็กต์ต่างๆ เข้าด้วยกัน
- F. **Command Button** เป็นปุ่มคำสั่งเพื่อใช้ในการสั่งงาน
- G. **Checkbox** เป็นปุ่มที่ใช้เลือกว่าต้องการหรือไม่
- H. **Option Button** ใช้เป็นตัวเลือกค่าใดค่าหนึ่งจากหลายๆค่า
- I. **Combo Box** ผู้ใช้สามารถเลือกตัวเลือกได้จากการกดปุ่ม Drop Down เพื่อแสดงทางเลือกต่างๆ มีความสามารถเหมือนกับ List Box และ Text Box ผสมกัน
- J. **List Box** ใช้แสดงตัวเลือกต่างๆ ในลักษณะของบรรทัดรายการโดยสามารถเลือกรายการใดรายการหนึ่งหรือว่าหลายรายการก็ได้
- K. **Horizontal Scroll Bar** เป็นแถบเลื่อนทางแนวนอน
- L. **Vertical Scroll Bar** เป็นแถบเลื่อนทางแนวตั้ง
- M. **Time** ใช้ในการควบคุมเวลา ในโปรแกรมที่มีเวลามาเกี่ยวข้อง
- N. **Drive List Box** ใช้ในการติดต่อเพิ่มข้อมูลในแบบของ List Box
- O. **Directory List Box** ใช้ในการติดต่อเพิ่มข้อมูลในแบบของ Directory ใช้ในการติดต่อเพิ่มข้อมูลในแบบของ
- P. **File List Box** ใช้ในการติดต่อเพิ่มข้อมูลในแบบของ File List Box
- Q. **Shape** ใช้สร้างรูปทรงต่างๆ ลงบนฟอร์ม
- R. **Line** ใช้วาดเส้นต่างๆ ลงบนฟอร์ม
- S. **Image** เป็นคอนโทรล ที่ใช้ควบคุมภาพเหมือนกับ Picture แต่ความสามารถจะน้อยกว่า
- T. **Data Control** ใช้เชื่อมต่อกับฐานข้อมูล
- U. **OLE** เป็นคอนโทรลที่นำเอาโปรแกรมสำเร็จรูปที่มีความสามารถ OLE เข้ามาใช้เป็นออปเจ็กต์ในโปรเจ็กต์

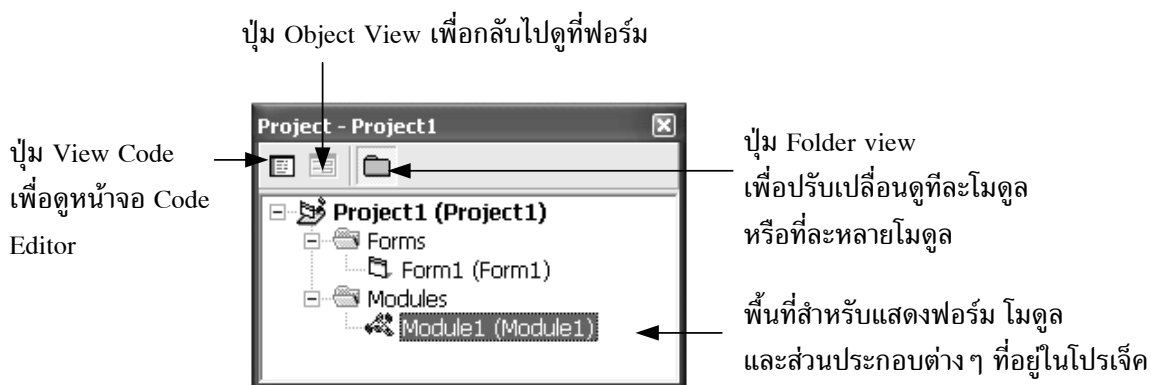
3) วินโดว์ Form



รูปที่ 2.7 วินโดว์ Form ใน Visual Basic 6.0

4) วินโดว์ Project Explorer

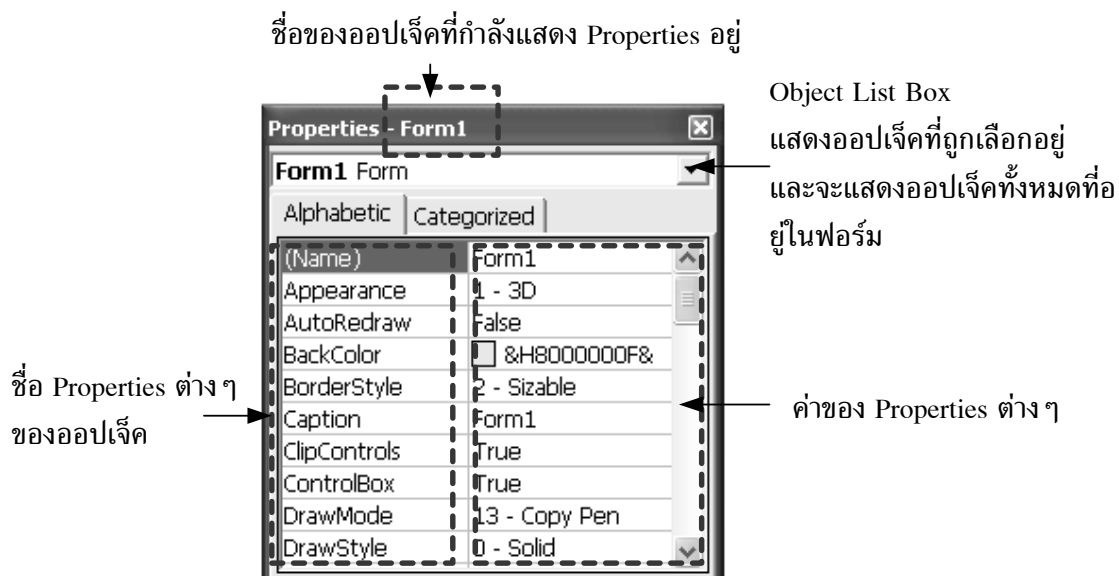
เป็นตัวแสดงรายละเอียดของโปรเจ็คต่างๆ ที่เราสร้างขึ้นมา ดังรูป



รูปที่ 2.8 วินโดว์ Project Explorer ใน Visual Basic 6.0

5) วินโดว์ Properties

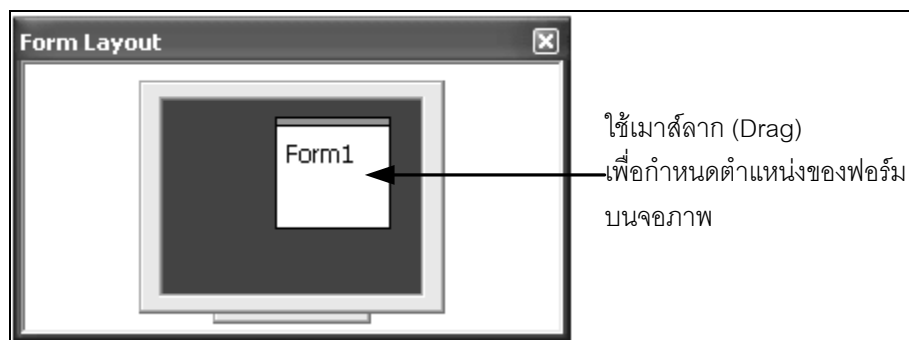
จะเป็นตัวแสดงคุณสมบัติทั้งหมด ของออปเจ็คที่กำลังถูกเลือกอยู่ ซึ่งในการแก้ไขหรือตั้งค่าคุณสมบัติ ทำได้โดยตรง ที่คุณสมบัติและค่า ซึ่งเราสามารถเรียกวินโดว์ Properties โดยเรียกได้จากเมนู View / Properties Windows หรือกด F4 ซึ่งวินโดว์ Properties ดูได้จาก ดังรูป



รูปที่ 2.9 วินโดว์ Properties ใน Visual Basic 6.0

6) วินโดว์ Form Layout

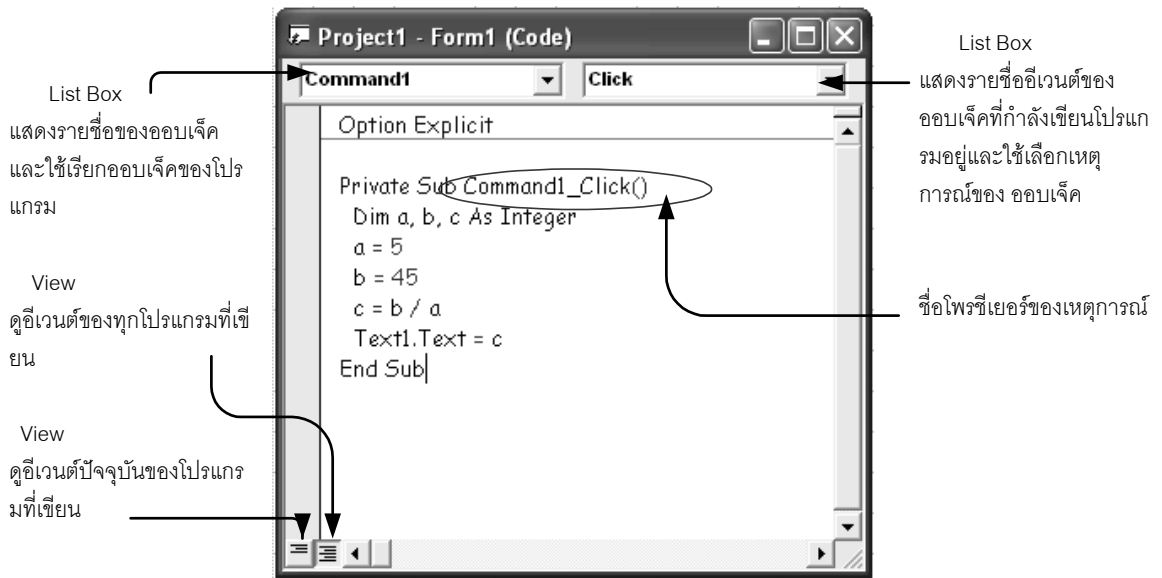
จะเป็นตัวแสดงตำแหน่งฟอร์มของโปรเจ็กต์ที่เรากำลังสร้างอยู่ ให้ดูบนจอภาพ เพื่อใช้กำหนดตำแหน่งขณะที่โปรแกรมทำงานจริงๆ ลักษณะของ Form layout Windows ดังรูป



รูปที่ 2.10 วินโดว์ Form Layout ใน Visual Basic 6.0

7) วินโดว์ Code Editor

เป็นเนื้อที่ใช้สำหรับเขียนโปรแกรม สามารถเรียกได้โดยใช้เมนู View/Code หรือดับเบิลคลิกที่ออบเจ็กต์ใดๆ บนฟอร์ม ซึ่งลักษณะของ Code Editor Windows ดังรูป



รูปที่ 2.11 วินโดว์ Code Editor ใน Visual Basic 6.0

7 ชนิดของข้อมูล ตัวแปร และค่าคงที่

1) บิต, ไบต์

บิต หมายถึง ข้อมูลที่เป็นค่าของตัวเลข '0','1' ซึ่งเป็นเลขฐานสอง จะเปรียบเสมือนกับ สวิตช์ที่มีเพียงสองสถานะ คือ เปิด และ ปิด เท่านั้น

ไบต์ หมายถึง ชุดของตัวเลขฐานสองแปดตัว เปรียบเสมือนกับสวิตช์แปดตัวเรียงกัน แต่ละชุดใช้แทนอักขระได้หนึ่งตัวอักษร

หน่วยของข้อมูลในคอมพิวเตอร์ดูได้จากตารางที่ 2.1

ตารางที่ 2.1 หน่วยของข้อมูลในคอมพิวเตอร์

จำนวน	เทียบเท่ากับ
8 บิต	1 ไบต์
1024 ไบต์	1 กิโลไบต์
1024 กิโลไบต์	1 เมกะไบต์
1024 เมกะไบต์	1 จิกะไบต์
1024 จิกะไบต์	1 เทอราไบต์

2) ตัวแปรในภาษาคอมพิวเตอร์

ตัวแปร (Variable) หมายถึงชื่อของหน่วยเก็บข้อมูล หรือตำแหน่งของหน่วยความจำของระบบคอมพิวเตอร์ ที่ใช้เก็บค่าต่างๆ ที่เปลี่ยนแปลงได้ระหว่างประมวลผล

ชนิดของตัวแปรใน Visual Basic มีดังนี้

ตารางที่ 2.2 ชนิดของตัวแปรใน Visual Basic

ชนิด	รายละเอียด	หน่วยความจำ
Byte	มีค่าตั้งแต่ 0 – 255	1 ไบต์
Integer	ใช้เก็บเลขจำนวนเต็ม ค่าตั้งแต่ -32,768 ถึง 32,767	2 ไบต์
Long	ใช้เก็บเลขจำนวนเต็ม ค่าตั้งแต่ -2,147,483,648 ถึง 2,147,483,647	4 ไบต์
Single	ใช้เก็บเลขจำนวนจริง ค่าตั้งแต่ -3.402823×10^{38} ถึง -1.0413×10^{-45} และค่าจริงตั้งแต่ 1.4013×10^{-45} ถึง 3.402823×10^{38}	4 ไบต์
Double	ใช้เก็บเลขจำนวนจริง ค่าตั้งแต่จำนวน $-1.79769313486232 \times 10^{308}$ ถึง $-4.94065645841247 \times 10^{-324}$ และค่าที่เป็นจำนวนจริงเริ่มตั้งแต่ค่า $4.94065645841247 \times 10^{-324}$ ถึง $1.79769313486232 \times 10^{308}$	8 ไบต์
Currency	ใช้เก็บเลขที่มีจำนวนตำแหน่งจำกัด ส่วนมากใช้เกี่ยวกับจำนวนเงิน มีค่าตั้งแต่ -922,337,203,685,477.5808 ถึง 922,337,203,685,477.5808	8 ไบต์
Boolean	ใช้เก็บค่าทางตรรกะ คือ จริง และ เท็จ	2 ไบต์
Date	ใช้เก็บค่าที่เป็นวันที่ และเวลา	8 ไบต์
Object	ใช้เก็บตัวแปรที่อ้างถึงออบเจ็คใดๆ	4 ไบต์
Variant	เป็นตัวแปรพิเศษของ Visual Basic สามารถเก็บค่าของตัวแปรได้ทุกแบบตามที่กล่าวมาข้างต้น	22 ไบต์

3) การประกาศตัวแปรใน Visual Basic

การประกาศตัวแปร คือการการที่เราบอกเครื่องคอมพิวเตอร์ให้เตรียมเนื้อที่ในหน่วยความจำสำหรับตัวแปรที่เราจะใช้ในการประมวลผลในโปรแกรม โดยการประกาศตัวแปรนั้นเราเรียกว่า Dim (Dimension) ซึ่งมีรูปแบบดังนี้

Dim ชื่อตัวแปร [As ชนิดของตัวแปร]

ค่าที่อยู่ภายในเครื่องหมาย [] จะเป็น Optional คือเราจะใส่หรือไม่ก็ได้ ถ้าไม่ใส่ ตัวแปรที่เราประกาศขึ้นจะเป็นตัวแปรชนิด Variant

ตัวอย่าง

Dim A As Integer ‘ ประกาศตัวแปรชื่อ A แบบ Integer
Dim B As String ‘ ประกาศตัวแปรชื่อ B แบบ String
Dim C As String * 10 ‘ ประกาศตัวแปรชื่อ C แบบ String มีความยาว 10 ตัวอักษร

ตัวแปรอีกแบบ คือ ตัวแปรแบบ ค่าคงที่ จะใช้ในการเก็บค่าคงที่ที่คาดว่าจะใช้บ่อยในโปรแกรม และค่าไม่มีการเปลี่ยนแปลง ซึ่งมีรูปแบบดังนี้

Constant ชื่อตัวแปร = ค่าคงที่

4) ขอบเขตของตัวแปร

ชนิดและขอบเขตของตัวแปรมีดังนี้

ตัวแปร Public จะมองเห็นและเรียกใช้ได้ทุกฟอร์มโมดูล หรือทุกโปรซีเจอร์ในโมดูล หรือฟอร์มที่ประกาศ การประกาศจะใช้คำว่า Public แทนคำว่า Dim ในการประกาศตัวแปร ต่อจากบรรทัด Option variable

ตัวแปร Private จะมองเห็นและเรียกใช้ได้ทุกโปรซีเจอร์ในโมดูล หรือฟอร์มที่เราประกาศเท่านั้น การประกาศจะใช้คำว่า Private แทนคำว่า Public ในการประกาศตัวแปรต่อจากบรรทัด Option variable

ตัวแปร Local จะมองเห็นและสามารถเรียกใช้ได้เฉพาะโปรซีเจอร์ที่ประกาศเท่านั้น ในการประกาศตัวแปรจะประกาศในโปรซีเจอร์ที่ใช้งาน

8 การใช้งาน Control ต่างๆ และการเขียนโปรแกรมควบคุม

1) คุณสมบัติพื้นฐานของคอนโทรล (Properties)

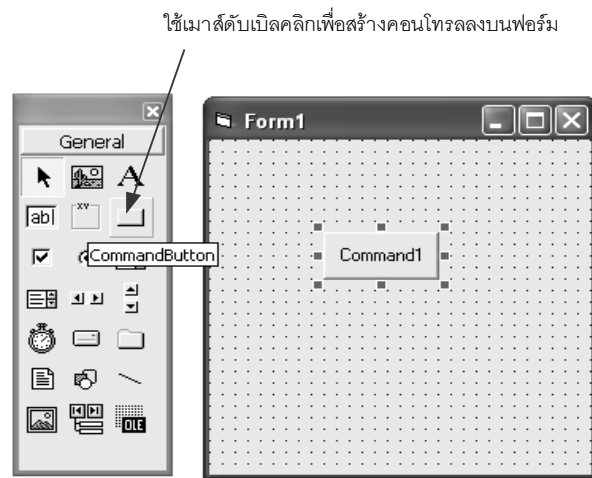
คุณสมบัติของคอนโทรล ใน Visual Basic ที่สำคัญแสดงดังตารางที่ 2.3

ตารางที่ 2.3 คุณสมบัติพื้นฐานของคอนโทรล (Properties)

ชื่อ Properties	รายละเอียด
Name	เป็นการกำหนดชื่อของคอนโทรล
Caption	เป็นข้อความอธิบายหรือป้ายบอกของคอนโทรลนั้น
Fore Color	ใช้กำหนดสีตัวอักษรของคอนโทรล
Back Color	ใช้กำหนดสี Background ของคอนโทรล
Height	ใช้กำหนดความสูงของคอนโทรล
Width	ใช้กำหนดความกว้างของคอนโทรล
Left	ใช้กำหนดตำแหน่งด้านซ้ายของคอนโทรล
Top	ใช้กำหนดตำแหน่งด้านบนของคอนโทรล
Visible	ใช้เป็นตัวกำหนดว่าจะให้คอนโทรลนั้นปรากฏหรือไม่
FontBuid, FontTltaliv, FontName, FontSize, FontStrikethru, FontUnderlin	ใช้กำหนดคุณลักษณะของตัวอักษร
Mouse Pointer	ใช้กำหนดลักษณะของเมาส์ เมื่อเลื่อนเข้ามาในคอนโทรล

2) การวาง Control ต่างๆ ลงบนฟอร์ม

ในการวางคอนโทรลลงบนฟอร์มทำได้โดยการดับเบิลคลิกที่รูปคอนโทรล หรือคลิกซ้ายหนึ่งครั้งแล้ววาดคอนโทรลลงบนฟอร์มตามตำแหน่งที่ต้องการจัดรูป



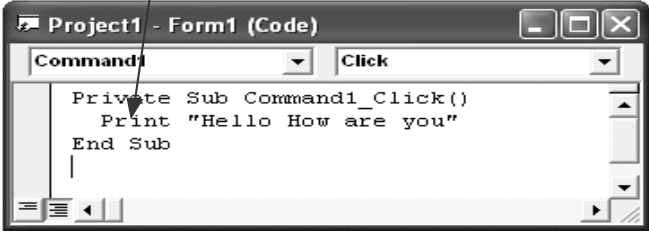
รูปที่ 2.12 การวาง Control ต่างๆ ลงบนฟอร์ม ใน Visual Basic 6.0

3) การใช้งานคอนโทรลต่างๆ

ในตัวอย่างในการใช้งานคอนโทรลเราจะยกตัวอย่างเฉพาะบางคอนโทรลที่จำเป็นและใช้งานบ่อยๆ เท่านั้น


 Command Button	เป็นปุ่มคำสั่งเพื่อใช้ในการสั่งงาน
<div style="text-align: center;">  </div> <p>รูปที่ 3.13 Command Button ใน Visual Basic 6.0</p> <p><i>ตัวอย่างการเขียนโปรแกรมควบคุม Command Button</i></p> <p>เมื่อเรากดปุ่ม Command Button จะแสดงคำว่า “Hello! How are you ?” ลงบนฟอร์ม</p> <p><u>ลำดับขั้นตอน</u></p> <ol style="list-style-type: none"> 1. วาด command Button ลงบนฟอร์ม 2. ดับเบิลคลิกที่ Command Button เพื่อเข้าสู่ Code Editor 3. เขียน Code Program ใน Code Editor ในส่วนของ Even Command1 Click() ดังนี้ 	

คำสั่งในการแสดงข้อความลงบนฟอร์ม




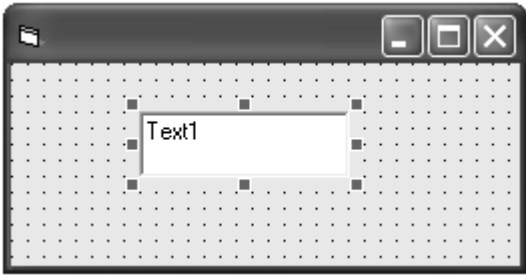
รูปที่ 2.14 Code Program ในส่วนของ Even Command1 Click() การควบคุม Command Button

4. กด F5 เพื่อรัน โปรแกรมแล้วคลิกที่ Command Button จะได้ผลดังนี้



รูปที่ 2.15 ผลการรัน โปรแกรม การควบคุม Command Button

 Text Box	เป็นคอนโทรลที่ใช้ในการรับข้อความ
--	----------------------------------

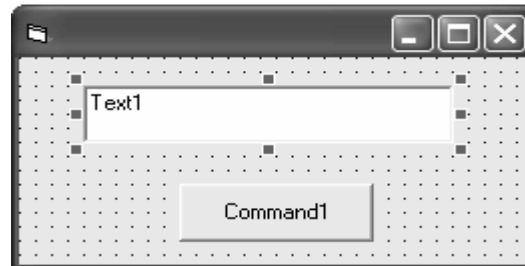


รูปที่ 2.16 Text Box ใน Visual Basic 6.0

ตัวอย่างการเขียนโปรแกรมควบคุม Text Box
 เมื่อเรากดปุ่ม Command Button จะแสดงคำว่า "Hello! How are you ?" ลงบน Text Box

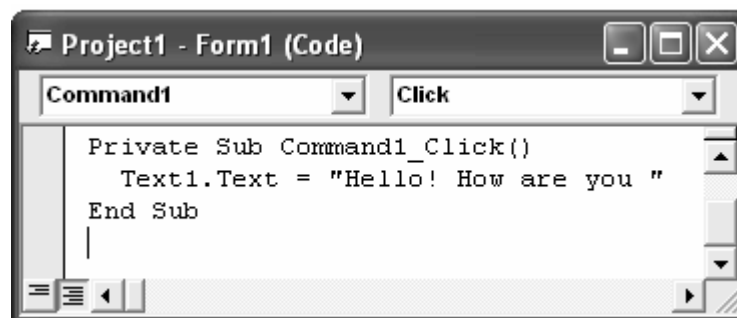
ลำดับขั้นตอน

1. วาด command Button และ Text Box ลงบนฟอร์มดังรูป



รูปที่ 2.17 การออกแบบฟอร์มสำหรับตัวอย่าง การควบคุม Text Box

2. เขียน Code Program ใน Code Editor ในส่วนของ Even Command1 Click() ดังนี้


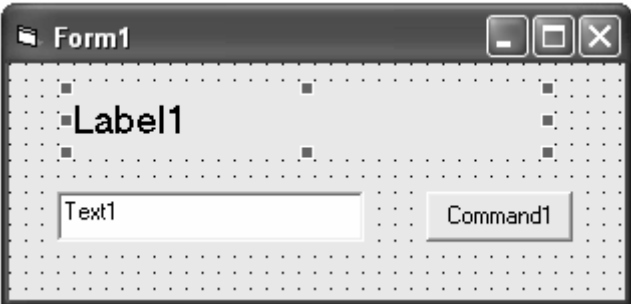


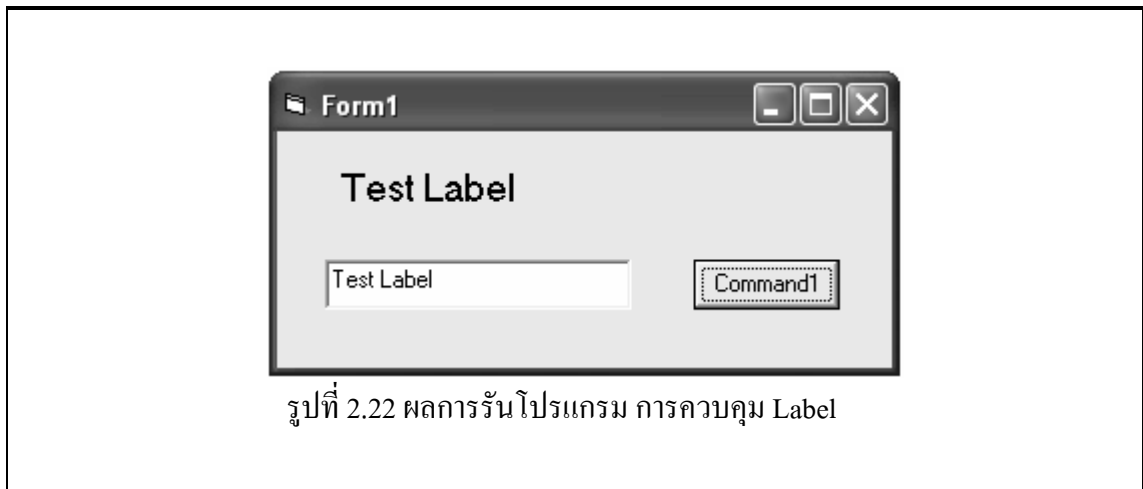
รูปที่ 2.18 Code Program ในส่วนของ Even Command1 Click() การควบคุม Text Box


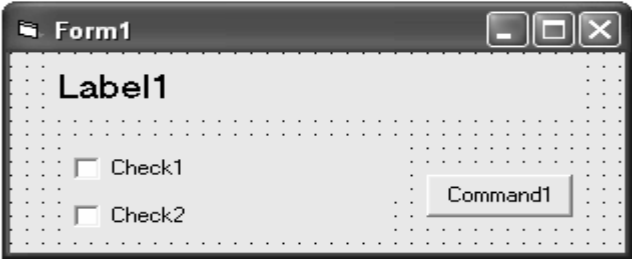
3. กด F5 เพื่อรัน โปรแกรมแล้วคลิกที่ Command Button จะได้ผลดังนี้



รูปที่ 2.19 ผลการรันโปรแกรม การควบคุม Text Box

<p>A</p> <p>Label</p>	<p>ใช้แสดงข้อความต่างๆ ลงบนฟอร์ม เมื่อลาก Label ลงบนฟอร์มแล้วจะได้ลักษณะดังรูป</p>		
<div style="text-align: center;">  <p>รูปที่ 2.20 Label ใน Visual Basic 6.0</p> </div> <p><i>ตัวอย่างการเขียนโปรแกรมควบคุม Label</i></p> <p>เมื่อเรากดปุ่ม Command Button โปรแกรมจะทำการรับค่าตัวอักษรที่เราพิมพ์ใน Text Box มาแสดงใน Label1</p> <p><u>ลำดับขั้นตอน</u></p> <ol style="list-style-type: none"> 1. วาด Label, Command Button และ Text Box ลงบนฟอร์มดังรูป <div style="text-align: center;">  <p>รูปที่ 2.21 การออกแบบฟอร์มสำหรับตัวอย่าง การควบคุม Label</p> </div> <ol style="list-style-type: none"> 2. เขียน Code Program ที่ Even Command1_Click() ดังข้างล่าง <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top;"> <pre>Private Sub Command1_Click() Label1.Caption = Text1.Text End Sub</pre> </td> <td style="width: 50%; vertical-align: top;"> <p>‘จะทำที่อีเวนต์นี้เมื่อเรากดคลิกที่ปุ่ม Command Button</p> <p>‘รับค่าจาก Text Box ไปแสดงที่ Caption ของ Label1</p> <p>‘จบการทำงาน</p> </td> </tr> </table> <ol style="list-style-type: none"> 3. กด F5 เพื่อรันโปรแกรม ทำการใส่ตัวอักษรที่ text box แล้วคลิกที่ Command Button จะได้ผลดังนี้ 		<pre>Private Sub Command1_Click() Label1.Caption = Text1.Text End Sub</pre>	<p>‘จะทำที่อีเวนต์นี้เมื่อเรากดคลิกที่ปุ่ม Command Button</p> <p>‘รับค่าจาก Text Box ไปแสดงที่ Caption ของ Label1</p> <p>‘จบการทำงาน</p>
<pre>Private Sub Command1_Click() Label1.Caption = Text1.Text End Sub</pre>	<p>‘จะทำที่อีเวนต์นี้เมื่อเรากดคลิกที่ปุ่ม Command Button</p> <p>‘รับค่าจาก Text Box ไปแสดงที่ Caption ของ Label1</p> <p>‘จบการทำงาน</p>		

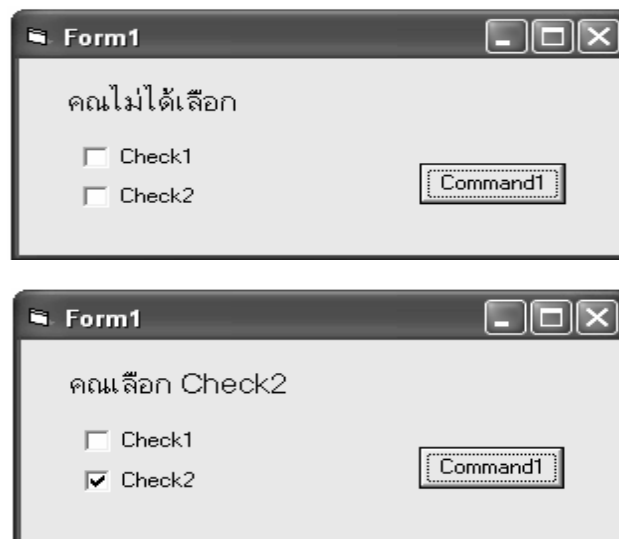


<input checked="" type="checkbox"/> Checkbox	เป็นปุ่มที่ใช้เลือกว่าต้องการหรือไม่ ลักษณะจะเป็นช่องให้เช็คหน้าข้อความ ดังรูป
<div style="text-align: center;">  <p>รูปที่ 2.23 Checkbox ใน Visual Basic 6.0</p> <p><i>ตัวอย่างการเขียนโปรแกรมควบคุม Check Box</i></p> <p>เมื่อเรากดปุ่ม Command Button โปรแกรมจะทำการรับค่าที่เช็คใน Check box แล้วแสดงผลว่าเราเช็คที่ CheckBox ได้บ้าง</p> <p><u>ลำดับขั้นตอน</u></p> <p>1. วาด Check box, Label และ command Button ลงบนฟอร์มดังรูป</p> <div style="text-align: center;">  </div> <p>รูปที่ 2.24 การออกแบบฟอร์มสำหรับตัวอย่าง การควบคุม Check Box</p> </div>	


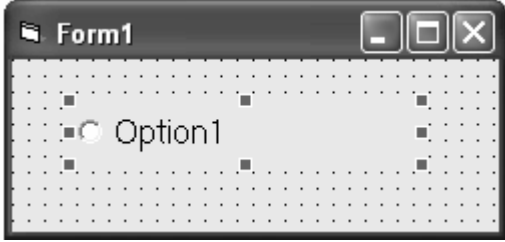

2. เขียน Code Program

```
Private Sub Command1_Click()
    If ((Check1.Value = 0) And (Check2.Value = 0)) Then
        Label1.Caption = "คุณไม่ได้เลือก"
    ElseIf ((Check1.Value = 1) And (Check2.Value = 0)) Then
        Label1.Caption = "คุณเลือก Check1"
    ElseIf ((Check1.Value = 0) And (Check2.Value = 1)) Then
        Label1.Caption = "คุณเลือก Check2"
    ElseIf ((Check1.Value = 1) And (Check2.Value = 1)) Then
        Label1.Caption = "คุณเลือก Check1 และ Check2"
    End If
End Sub
```

3. กด F5 เพื่อรัน โปรแกรมแล้วคลิกที่ Command Button จะได้ผลดังนี้



รูปที่ 2.25 ผลการรันโปรแกรม การควบคุม Check Box

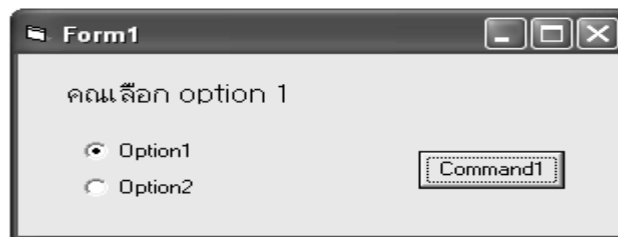
 <p>Option Button</p>	<p>ใช้เป็นตัวเลือกค่าใดค่าหนึ่งจากหลายๆค่า ดังรูป</p>
<div style="text-align: center;">  <p>รูปที่ 2.26 Option Button ใน Visual Basic 6.0</p> <p><i>ตัวอย่างการเขียน โปรแกรมควบคุม Option Button</i></p> <p>เมื่อเรากดปุ่ม Command Button โปรแกรมจะทำการรับค่าที่เช็คใน Option Button แล้วแสดงผลว่าเราได้เลือกที่ Option ใด ลงบน Label</p> <p><u>ลำดับขั้นตอน</u></p> <ol style="list-style-type: none"> 1. วาด Option Button, Label และ command Button ลงบนฟอร์มดังรูป <div style="text-align: center;">  <p>รูปที่ 2.27 การออกแบบฟอร์มสำหรับตัวอย่าง การควบคุม Option Button</p> <ol style="list-style-type: none"> 2. เขียน Code Program ดังตารางข้างล่าง <pre> Private Sub Command1_Click() If ((Option1.Value = False) And (Option2.Value = False)) Then Label1.Caption = "ไม่ได้เลือก" ElseIf ((Option1.Value = True) And (Option2.Value = False)) Then Label1.Caption = "คุณเลือก option 1" End If End Sub </pre> </div> </div>	

```


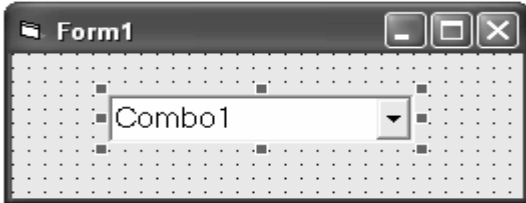
ElseIf ((Option1.Value = False) And (Option2.Value = True)) Then
    Label1.Caption = "" คุณเลือก option 2"
End If
End Sub

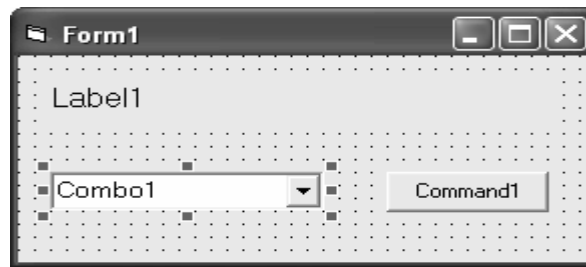
```

3. กด F5 เพื่อรันโปรแกรมแล้วคลิกที่ Command Button จะได้ผลดังนี้



รูปที่ 2.28 ผลการรันโปรแกรม การควบคุม Option Button

 Combo Box	<p>ผู้ใช้สามารถเลือกตัวเลือกได้จากการกดปุ่ม Drop Down เพื่อแสดงทางเลือกต่างๆ ดังรูป</p>
<div style="text-align: center;">  <p>รูปที่ 2.29 Combo Box ใน Visual Basic 6.0</p> </div> <p><i>ตัวอย่างการเขียน โปรแกรมควบคุม Combo Box</i></p> <p>เมื่อเรากดปุ่ม Command Button โปรแกรมจะทำการรับค่าที่เช็คใน Combo Box เพื่อใช้เป็นตัวกำหนดขนาดของตัวอักษรของ Label</p> <p><u>ลำดับขั้นตอน</u></p> <ol style="list-style-type: none"> 1. วัด Combo Box, Label และ command Button ลงบนฟอร์มดังรูป 	



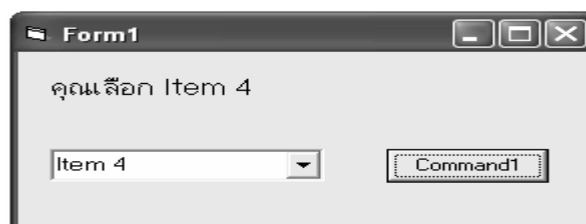
รูปที่ 2.30 การออกแบบฟอร์มสำหรับตัวอย่าง การควบคุม Combo Box

2.เขียน Code Program ดังรูปข้างล่าง


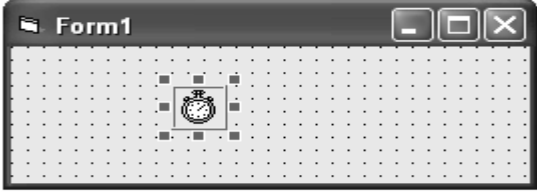

```
Private Sub Command1_Click()
    Label1.Caption = "๘0³àÅ×Í; " & Combo1.Text
End Sub

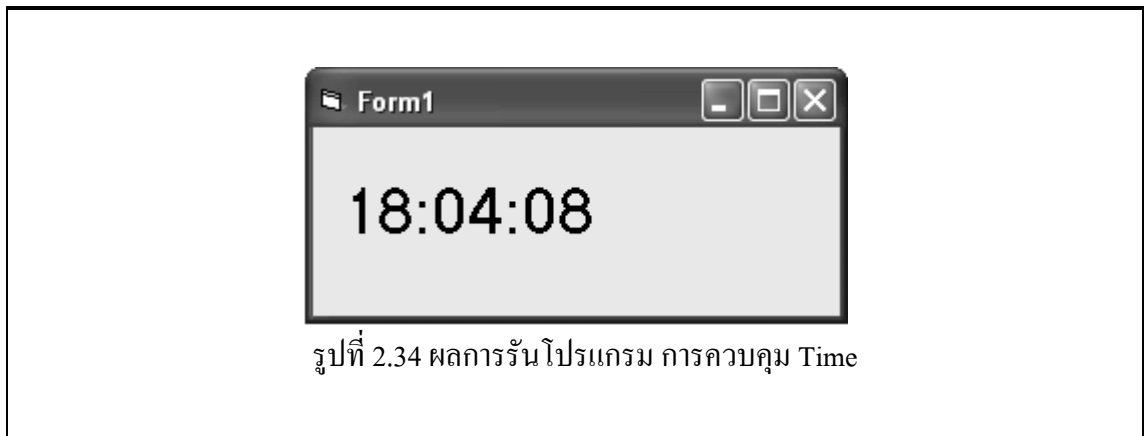
Private Sub Form_Load()
    For i = 1 To 5
        Combo1.AddItem "Item " & i
    Next
End Sub
```

3. กด F5 เพื่อรัน โปรแกรมแล้วคลิกที่ Command Button จะได้ผลดังนี้



รูปที่ 2.31 ผลการรัน โปรแกรม การควบคุม Combo Box

 Time	ใช้ในการควบคุมเวลา ในโปรแกรมที่มีเวลามาเกี่ยวข้องกับ Time จะมีลักษณะดังรูป
<div style="text-align: center;">  <p>รูปที่ 2.32 Time ใน Visual Basic 6.0</p> <p>ตัวอย่างการเขียนโปรแกรมควบคุม Time เป็นโปรแกรมที่แสดงเวลาของปัจจุบัน ลงบน Label ซึ่งในหนึ่งวินาทีจะเรียกเวลาขึ้นมา แสดงใหม่</p> <p><u>ลำดับขั้นตอน</u></p> <ol style="list-style-type: none"> 1. วัด Time และ Label ลงบนฟอร์มดังรูป <div style="text-align: center;">  </div> <p>รูปที่ 2.33 การออกแบบฟอร์มสำหรับตัวอย่าง การควบคุม Time</p> <ol style="list-style-type: none"> 2. ดับเบิลคลิกที่ Timer1 แล้วเขียน Code Program ข้างล่าง <pre>Private Sub Timer1_Timer() ‘ จะทำงานทุกๆ 1 วินาที Label1.Caption = Time() ‘ เป็นการเรียกเวลาจากฟังก์ชัน Time() มาแสดง End Sub ‘ ที่ Label1</pre> 4. กด F5 เพื่อรันโปรแกรม จะได้ผลดังนี้ </div>	



9 การควบคุมเมาส์ คีย์บอร์ด และ MsgBox

ต่อไปเราจะกล่าวถึงการเขียน โปรแกรม ที่เกี่ยวเหตุการณ์ต่างๆ ที่เกิดจากการใช้เมาส์และคีย์บอร์ด และรู้จักฟังก์ชัน MsgBox ที่ใช้ในการแสดงกรอบข้อความมาตรฐาน ของวินโดว์ ให้กับผู้ใช้

1) การควบคุมเมาส์

เมาส์เป็นอุปกรณ์ที่ชี้ตำแหน่งที่มีความสำคัญในการทำงานของระบบวินโดว์ เราจะได้เรียนรู้การใช้เมาส์ (Mouse) มาควบคุมออบเจกต์ต่างๆ ใน Visual Basic

เหตุการณ์ของเมาส์ใน Visual Basic

ตารางที่ 2.4 เหตุการณ์ของเมาส์ใน Visual Basic

เหตุการณ์	รายละเอียด
MouseDown	จะเกิดขึ้นเมื่อมีการกดปุ่มของเมาส์
MouseUp	จะเกิดขึ้นเมื่อมีการปล่อยปุ่มกดของเมาส์
MouseMove	จะเกิดขึ้นเมื่อมีการเคลื่อนเมาส์

รูปแบบและพารามิเตอร์ ที่เกี่ยวข้องกับเมาส์

Private Sub ชื่อออบเจ็กต์_MouseDown(Button As Integer, Shift As Integer, _
X As Single, Y As Single)

Private Sub ชื่อออบเจ็กต์_MouseUp(Button As Integer, Shift As Integer, _
X As Single, Y As Single)

Private Sub ชื่อออบเจ็กต์_MouseMove(Button As Integer, Shift As Integer, _
X As Single, Y As Single)

Button เป็นตัวรับค่าของปุ่มเมาส์ที่กด ซึ่งเป็นตัวแปรแบบ Integer เมื่อเรากดปุ่มของเมาส์ จะมีค่าต่างๆ ดังนี้

ตารางที่ 2.5 ค่า Button ของปุ่มเมาส์ที่กด

ปุ่มที่กด	ค่าของ Button
ปุ่มซ้าย	1
ปุ่มขวา	2
ปุ่มซ้าย+ปุ่มขวา	3
ปุ่มกลาง	4
ทั้งสามปุ่ม	7

Shift เป็นตัวแสดงสถานะของคีย์ Alt, Ctrl และ Shift เป็นแบบ Integer เหมือนกับ Button มีการกำหนดค่าไว้เช่นเดียวกับเมาส์ดังนี้

ตารางที่ 2.6 ค่า Button แสดงสถานะของคีย์ Alt, Ctrl และ Shift

ปุ่มที่กด	ค่าของ Button
ปุ่ม Shift	1
ปุ่ม Ctrl	2
ปุ่ม Alt	4
ทั้งสามปุ่ม	7

X และ Y เป็นตัวบอกตำแหน่งเคอร์เซอร์ของเมาส์ในแกน X, Y

2) การควบคุมคีย์บอร์ด

ในส่วนนี้เราจะกล่าวถึงการตรวจสอบการกดคีย์ต่างๆ ของคีย์บอร์ด รวมทั้งเหตุการณ์ต่างๆ ที่เกี่ยวข้อง เข้ามาใช้งานหรือเขียน โปรแกรม

เหตุการณ์ที่เกี่ยวข้องกับคีย์บอร์ดมีอยู่หลักๆ 3 เหตุการณ์คือ

ตารางที่ 2.7 เหตุการณ์ที่เกี่ยวข้องกับคีย์บอร์ดใน Visual Basic

เหตุการณ์	รายละเอียด
KeyDown	เป็นเหตุการณ์ที่เกิดขึ้นเมื่อมีการกดปุ่มใดๆ บนคีย์บอร์ด
KeyUp	เหตุการณ์ที่เกิดขึ้นเมื่อมีการปล่อยปุ่มกดใดๆ บนคีย์บอร์ด
KeyPress	เป็นเหตุการณ์ที่เกิดขึ้นเมื่อเราได้มีการกดปุ่มใดๆ ลงบนคีย์บอร์ด โดยรวมเหตุการณ์ของ KeyDown และ KeyUp แต่จะไม่ตรวจสอบการกดคีย์พิเศษ เช่น Ctrl, Shift และ Alt

รูปแบบและพารามิเตอร์ที่เกี่ยวข้องกับคีย์บอร์ด

Private Sub ชื่อออบเจ็กต์_KeyDown(KeyCode As Integer, Shift As Integer)

Private Sub ชื่อออบเจ็กต์_KeyUp(KeyCode As Integer, Shift As Integer)

Private Sub ชื่อออบเจ็กต์_KeyPress(KeyAscii As Integer)

Button เป็นตัวรับค่าของปุ่มเมาส์ที่กด ซึ่งเป็นตัวแปรแบบ Integer เมื่อเรากดปุ่มของเมาส์จะมีค่าต่างๆ ดังนี้

ตารางที่ 2.8 ค่า Button ของปุ่มเมาส์ที่กด

ปุ่มที่กด	ค่าของ Button
ปุ่มซ้าย	1
ปุ่มขวา	2
ปุ่มซ้าย+ปุ่มขวา	3
ปุ่มกลาง	4
ทั้งสามปุ่ม	7

Shift เป็นตัวแสดงสถานะของคีย์ Alt, Ctrl และ Shift เป็นแบบ Integer เหมือนกับ Button มีการกำหนดค่าไว้เช่นเดียวกับเมาส์ดังนี้

ตารางที่ 2.9 ค่า Button แสดงสถานะของคีย์ Alt, Ctrl และ Shift

ปุ่มที่กด	ค่าของ Button
ปุ่ม Shift	1
ปุ่มCtrl	2
ปุ่ม Alt	4
ทั้งสามปุ่ม	7

X และ **Y** เป็นตัวบอกตำแหน่งเคอร์เซอร์ของเมาส์ในแกน X, Y

3) MsgBox

MsgBox ที่ใช้ในการแสดงกรอบข้อความมาตรฐานของวินโดว ให้กับผู้ใช้ ซึ่งมีรูปแบบดังนี้





MsgBox (prompt [,Buttons] [,Title])

Prompt คือข้อความที่ต้องการให้แสดง




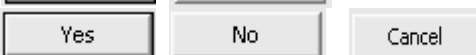
Buttons ใช้ในการกำหนดปุ่มและไอคอนที่ปรากฏใน MsgBox

Title ใช้กำหนดข้อความบนแถบ

ค่าคงที่ในการกำหนดไอคอนบน MsgBox

vbCritical	ไอคอนที่ปรากฏคือ	
vbExclamation	ไอคอนที่ปรากฏคือ	
vbInformation	ไอคอนที่ปรากฏคือ	
vbQuestion	ไอคอนที่ปรากฏคือ	

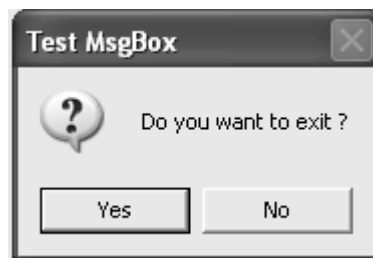
ค่าคงที่ในการกำหนดปุ่มบน MsgBox

vbOkOnly	ปุ่มที่ปรากฏคือ	
vbOkCancel	ปุ่มที่ปรากฏคือ	
vbYesNo	ปุ่มที่ปรากฏคือ	
vbYesNoCancel	ปุ่มที่ปรากฏคือ	

ตัวอย่างการใช้งาน MsgBox

Code = MsgBox("Do you want to exit ? ", vbYesNo + vbQuestion, "Test MsgBox")

ผลที่ได้จากตัวอย่างแสดงดังรูป



รูปที่ 2.35 ผลการรันโปรแกรม ตัวอย่างการใช้งาน MsgBox

จากตัวอย่างเป็นการใช้ตัวแปร Code ในการรับค่าจาก MsgBox เมื่อเรากดปุ่ม Yes ค่าที่ได้จาก Code คือ vbYes ถ้ากดปุ่ม No ค่าที่ได้ก็คือ vbNo ซึ่งเราสามารถนำค่าที่ได้ไปเป็นตัวตรวจสอบเพื่อใช้ในโปรแกรมได้

10 ประโยคคำสั่ง (Statement)

Statement หรือประโยคคำสั่ง หมายถึง คำสั่งในโปรแกรมที่เราเขียนขึ้น เพื่อใช้ควบคุมการทำงานของโปรแกรม โดยสามารถแบ่งออกเป็นสามประเภทใหญ่ๆ คือ

1) ประโยคกำหนดค่า

ประโยคกำหนดค่า (Assignment Statement) คือการเขียนคำสั่งที่ป้อนค่าหรือใส่ค่าให้กับตัวแปรด้วยนิพจน์ต่างๆ ตัวดำเนินการหลักที่ใช้ในประโยคที่ใช้กำหนดค่าของ Visual Basic คือ เครื่องหมาย “=”

ตัวอย่างเช่น	$A = 100$	‘กำหนดให้ A เท่ากับ 100
	$Sum = A + B + C$	‘กำหนดให้ Sum เท่ากับ $A + B + C$
	$X = X + 1$	‘กำหนดให้ X เท่ากับ $X + 1$

2) ประโยคเงื่อนไข

ประโยคเงื่อนไข (Condition Statement) จะเป็นประโยคคำสั่งในการสร้างเงื่อนไข เพื่อใช้ในการตัดสินใจตามเงื่อนไขที่ได้กำหนดไว้ คำสั่งประเภทนี้ได้แก่ IF-THEN-ELSE และ SELECT-CASE เป็นต้น

ก) คำสั่ง IF-THEN-ELSE

เป็นคำสั่งที่ใช้เมื่อต้องการให้คอมพิวเตอร์ตัดสินใจว่าจะไปทำงานตรงส่วนไหน ซึ่งมีรูปแบบดังนี้

```
IF ( เงื่อนไข ) THEN
.....
ELSE
.....
END IF
```

ข) คำสั่ง SELECT-CASE

เป็นคำสั่งที่ใช้ในการเลือกทำ จะมีการทำงานคล้ายกับ IF-THEN-ELSE แต่ Select Case จะเหมาะสำหรับตรวจสอบค่าตัวแปรเดียว รูปแบบคำสั่ง SELECT-CASE มีรูปแบบดังนี้

SELECT CASE ตัวแปร

Case (ค่าเปรียบเทียบ)

.....

Case (ค่าเปรียบเทียบ)

.....

Case (ค่าเปรียบเทียบ)

.....

END SELECT

3) ประโยควนรอบ

ประโยควนรอบ หรือที่เราเรียกว่าการวนลูป (Iteration Statement) ซึ่งคำสั่งประเภทนี้ได้แก่ FOR-NEXT และ DO-WHILE เป็นต้น

ก) คำสั่ง FOR-NEXT

รูปแบบคำสั่ง FOR-NEXT

FOR ตัวแปร = ค่าเริ่มต้น TO ค่าสิ้นสุด [ค่าที่เพิ่มขึ้นแต่ละรอบ]

ชุดคำสั่ง

.....

NEXT

ข) คำสั่ง DO-WHILE-LOOP

รูปแบบคำสั่ง DO-WHILE-LOOP

DO WHILE (เงื่อนไข)

ชุดคำสั่ง

.....

LOOP

11 ฟังก์ชันต่างๆ ใน Visual Basic

ในบทนี้จะเป็นการเรียนรู้ฟังก์ชันต่าง ๆ ที่ทำงานกับตัวแปรแบบ String และตัวเลขที่ Visual Basic ได้จัดเตรียมมาให้ใช้มากมาย โดยที่เราไม่ต้องเสียเวลาไปพัฒนาหรือเขียนฟังก์ชันนี้ขึ้นมาใช้เองฟังก์ชันการทำงานเกี่ยวกับตัวเลข (Math Function) และ String ในบทนี้จะเป็นฟังก์ชันที่เรามักจะใช้บ่อย ๆ ในการเขียนโปรแกรม การทำความเข้าใจและใช้งานคำสั่งหรือฟังก์ชันเหล่านี้จะเป็นพื้นฐานที่ช่วยในการเขียนโปรแกรมแก้ปัญหาต่าง ๆ เป็นอย่างมาก

1) ฟังก์ชันที่ใช้ในการปัดเศษ

ก) ฟังก์ชัน Round ()

รูปแบบ Round(expression [,numdecimalplaces])

ใช้ในการปัดเศษ ให้เท่ากับจำนวนทศนิยมตำแหน่งที่ numdecimalplaces โดยถ้าเลขตัวท้ายมากกว่าหรือเท่ากับ 5 จะปัดเศษขึ้น ถ้าน้อยกว่า 5 จะปัดเศษลง

ตัวอย่าง

Dim I As Variant, result As Variant

i = 1.5467

result = Round (i, 0) ‘ผลลัพธ์ = 2

result = Round (i, 1) ‘ผลลัพธ์ = 1.5

result = Round (i, 1.55) ‘ผลลัพธ์ = 1.55

result = Round (i, 3) ‘ผลลัพธ์ = 1.547

ข) ฟังก์ชัน Into (Number) และฟังก์ชัน Fix (Number)

ทั้งฟังก์ชัน Into (Number) และฟังก์ชัน Fix (Number) ใช้ในการหาค่าส่วนที่เป็นจำนวนเต็มของตัวเลข Number โดยตัดค่าทศนิยมของตัวเลขทิ้งไป ฟังก์ชัน Into กับ Fix จะทำงานต่างในกรณีที่ค่า ตัวเลขเป็นค่าลบ (Negative) กล่าวคือ Into จะได้ค่าจำนวนเต็มตัวแรกที่น้อยกว่าหรือเท่ากับ Number แต่ Fix จะได้ค่าจำนวนเต็มตัวแรกที่มากกว่าหรือเท่ากับ Number

ตัวอย่าง

Dim MyNumber	MyNumber = Into (99.8)	'Returns 99.
MyNumber = Fix	(99.2)	'ผลลัพธ์ 99
MyNumber = Into	(-99.8)	'ผลลัพธ์ -100
MyNumber = Fix	(-99.8)	'ผลลัพธ์ -99
MyNumber = Into	(-99.2)	'ผลลัพธ์ -100
MyNumber = Fix	(-99.2)	'ผลลัพธ์ -99

2) ฟังก์ชันเกี่ยวกับค่ายกกำลังและ Exponential

ก) ฟังก์ชัน Exp (Number)

จะได้ค่า e ยกกำลัง Number โดย $e=2.71828$

ข) ฟังก์ชัน Log (Number)

จะได้ค่า In (Number) โดย In คือ Log ฐาน e (2.71828) เช่น $\text{Log}(10) = 2.30258$ เป็นต้น การเปลี่ยนค่า Log ไปเป็นฐานอื่นสามารถทำได้โดยใช้สูตร เป็นการเปลี่ยนค่าเป็น Log ของ x ในฐาน N $\text{Log}_N(X) = \text{Log}(X) / \text{Log}(N)$

ค) ฟังก์ชัน Sqr (Number)

สำหรับหาค่ารากที่ 2 ของ Number

ตัวอย่าง

Dim MySqr		
MySqr = Sqr	(9)	'ผลลัพธ์ 3
MySqr = Sqr	(7)	'ผลลัพธ์ 2.64575131106459
MySqr = Sqr	(0)	'ผลลัพธ์ 0
MySqr = Sqr	(-5)	'เกิด error

3) ฟังก์ชันในการสุ่มตัวเลข

ก) ฟังก์ชัน Rnd

จะได้ค่าสุ่มตัวเลขตั้งแต่ 0 ถึงค่าน้อยกว่า 1 (0.999...) ที่ขึ้นกับเลขฐานถ้าเป็นจำนวนเต็ม ก็จะได้ค่าเต็มตลอดจึงต้องใช้คำสั่ง Randomize ก่อนการใช้งานใช้ฟังก์ชัน Rnd เพื่อให้ค่าที่สุ่มได้มีการเปลี่ยนแปลงฐานของเลขก่อน

ตัวอย่าง

```
Dim MyValue, i
Randomize 'สร้างตารางเลขสุ่มใหม่
For i = 1 to 1000
    MyValue = Into((6*Rnd) + 1) 'สุ่มตัวเลขตั้งแต่ 1 ถึง 6
    Print i
Next i
```

4) ฟังก์ชันในการหาค่าสัมบูรณ์และหาเครื่องหมายของตัวเลข

ก) ฟังก์ชัน Abs(Number)

หาค่าสัมบูรณ์ของตัวเลข Number เช่น $Abs(-1) = 1$, $Abs(2) = 2$ เป็นต้น

ข) ฟังก์ชัน Sgn (Number)

ใช้ในการตรวจสอบหาเครื่องหมายบวกหรือลบของ Number โดย

ถ้า Number มากกว่า 0 'จะได้ค่ากลับมาเป็น 1

ถ้าค่า Number = 0 'จะได้ค่ากลับมาเป็น 0

ถ้า Number น้อยกว่า 0 'จะได้ค่ากลับมาเป็น -1

ตัวอย่าง

```
Dim MyVar1, MyVar2, MyVar3, Mysign
MyVar1 = 10: MyVar2 = -5.4 : MyVar3 = 0
MySign = Sgn(MyVar1) 'ผลลัพธ์ 1
MySign = Sgn(MyVar2) 'ผลลัพธ์ -1
MySign = Sgn(MyVar3) 'ผลลัพธ์ 0
```

5) ฟังก์ชันในการแปลงเลขฐาน

ก) ฟังก์ชัน Hex (Number)

ใช้ในการแปลงเลขฐาน 10 เป็นสตริงในรูปแบบของเลขฐาน 16 ค่าที่ส่งกลับมาจากฟังก์ชัน Hex จะเป็นค่าตัวแปรแบบ String สำหรับค่าคงที่ใน Visual Basic เราสามารถกำหนดค่าตัวเลขที่เป็นเลขฐาน 16 ได้โดยใช้ตัวอักษร &H นำหน้า

ตัวอย่าง

```
Dim X As Variant
```

```
X = &H1A      'ค่า จะเท่ากับ 1 ฐาน 16 ซึ่งเท่ากับ 26 ในฐาน 10
```

```
Dim MyHex
```

```
MyHex = Hex(5)      'ผลลัพธ์ "5"
```

```
MyHex = Hex(10)     'ผลลัพธ์ "A"
```

```
MyHex = Hex(459)    'ผลลัพธ์ "1CB"
```

ข) ฟังก์ชัน Oct (Number)

ใช้ในการแปลงเลขฐาน 10 (Decimal) เป็นสตริงของเลขฐาน 8 (Octadecimal) สำหรับค่าคงที่ใน Visual Basic เราสามารถกำหนดค่าตัวเลขที่เป็นเลขฐาน 8 ได้โดยใช้ตัวอักษร &O นำหน้า

ตัวอย่าง

```
Dim X As Variant
```

```
X = &O71      'ค่า X จะเท่ากับ 71 ฐาน 8 ซึ่งเท่ากับ 57 ฐาน 10
```

```
Dim MyOct
```

```
MyOct = Oct(4)     'ผลลัพธ์ 4
```

```
MyOct = Oct(8)     'ผลลัพธ์ 10
```

```
MyOct = Oct(459)   'ผลลัพธ์ 713
```

6) ฟังก์ชันทางตรีโกณมิติ

ก) ฟังก์ชัน Sin (Number)

แสดงค่า Sine ของมุมที่กำหนดใน Number (ค่าของมุมจะมีหน่วยเป็น Radian โดยสามารถแปลงองศาเป็น Radian ได้จากสูตร

$$\text{Degree} = \text{Radian} * 180/\text{Pi} \text{ โดย } \text{Pi} = 3.141592654)$$

ข) ฟังก์ชัน Cos (Number)

แสดงค่า Cos ของมุมที่กำหนดใน Number

ค) ฟังก์ชัน Tan (Number)

แสดงค่า Tan ของมุมที่กำหนดใน Number