2.2.3 ฟังชั่นทางคณิตศาสตร์

2.2.3.1 abs(k) ใช้สำหรับแปรงค่าตัวเลขให้เป็นค่า absolute

ตัวอย่าง

k= abs(-10)

จะได้ k = 10

2.2.3.2 atn(k) ใช้สำหรับหาค่า arctangent

ตัวอย่าง

k=atn(pi/2) pi = 3.14

จะได้ k= 1

2.2.3.3 cos(k) ใช้สำหรับหาค่า cosine เมื่อ k เป็นองศา radians

ตัวอย่าง

k=cos(pi/6) pi=3.14

จะได้ k= 0.866

2.2.3.4 sin(k) ใช้สำหรับหาค่า sine เมื่อ k เป็นองศา radians

ตัวอย่าง

k = sin(pi/2) pi=3.14

จะได้ k = 1

2.2.3.5 tan(k) ใช้สำหรับหาค่า tan เมื่อ k เป็นองศา radians

ตัวอย่าง

k=tan(1.3)

จะได้ k= 3.6

2.2.3.6 Rnd ใช้สำหรับสุ่มค่าตัวเลข

ตัวอย่าง ต้องการให้มีการสุ่มตัวเลขระหว่าง 1 - 10

k = int((10 x Rnd)+1)

k จะมีค่าระหว่าง 1 -10

2.2.3.7 sgn ใช้สำหรับเช็กเครื่องหมายทางคณิตศาสตร์

ตัวอย่าง

k=-10

sgn(k)= -1

n=10 ; sgn(n) = 1

2.2.3.8 sqrใช้สำหรับหาค่า square ทางคณิตศาสตร์

ตัวอย่าง

k = sqr(9)

จะได้ k=3

2.2.3.9 int(k) และ fix(k) ใช้สำหรับแปลงตัวเลขที่มีจุดทศนิยมเป็นเลขจำนวนเต็ม

ตัวอย่างที่ 1

k=3.0009

b =int(k)

จะได้ b = 3

ตัวอย่างที่ 2

k=-3.5

a=fix(k) จะได้ a=-4

b=int(k) จะได้ b=-3

2.2.3.10 Hex ใช้สำหรับแปลงค่าตัวเลขให้เป็นเลขฐานสิบหก

ตัวอย่าง

k=255: a=Hex(k) จะได้ a = FF

2.2.3.11 Oct ใช้สำหรับแปลงค่าตัวเลขให้เป็นเลขฐานแปด

ตัวอย่าง

k=255: a=Oct(k) จะได้ a =377

2.2.3.12 str ใช้สำหรับแปลงตัวเลขให้เป็นตัวอักษร

ตัวอย่าง

k=100 : a=str(k) จะได้ a="100"

2.2.3.13 val ใช้สำหรับแปลงตัวอักษรเป็นตัวเลข

ตัวอย่างที่

a$="ABC123" : K=VAL(a$) จะได้ k=123

2.2.4 ฟังชั่นที่ใช้งานทางด้านวันและเวลา

2.2.4.1 date ใช้สำหรับบอก เดือน/วันที่/ปี ปัจจุบัน

ตัวอย่าง

a$= date จะได้ a$ = 03/17/05

2.2.4.2 day ใช้สำหรับบอก วันที่ จาก เดือน/วัน/ปี ปัจจุบัน

ตัวอย่าง

a$ = day(date) จะได้ a$="17"

2.2.4.3 month ใช้สำหรับบอก เดือน จาก เดือน/วัน/ปี ปัจจุบัน

ตัวอย่าง

a$ = month(date) จะได้ a$ = "3"

2.2.4.4 year ใช้สำหรับบอก ปี จาก เดือน/วัน/ปี ปัจจุบัน

ตัวอย่าง

a$=year(date) จะได้ a$= "05"

2.2.4.5 time ใช้สำหรับบอกเวลาปัจจุบัน

ตัวอย่าง

a$=time จะได้ a$ ="2:48:45 PM"

2.2.4.6 hour ใช้บอก ชั่วโมง จากเวลาปัจจุบัน

ตัวอย่าง

a$=hour(time) จะได้ a$= "14"

2.2.4.7 MINUTE ใช้บอก นาที จากเวลาปัจจุบัน

ตัวอย่าง

a$=minute(time) จะได้ a$="48"

2.2.4.8 second ใช้บอกวินาที จากเวลาปัจจุบัน

ตัวอย่าง

a$=second(time) จะได้ a$="22"

3

User Defined Procedure & User Defined Function
และ ชนิดของตัวแปร

3.3. User Defined Procedure

ปกติ visual basic ได้แบ่งโปรแกรมออกเป็นโปรแกรมย่อย ๆ ตามเหตุการณ์ ต่าง ๆ ที่เกิดขึ้นกับวัตถุต่าง ๆ ใน project เช่น sub command_click, sub form_load เป็นต้น แต่บางครั้งเราอาจมีความจำเป็นที่ต้องสร้างโปรแกรมย่อยเพื่อความสดวกในการทำงานของเรา ซึ่งต่อไปนี้จะเป็นรูปแบบการสร้างโปรแกรมย่อย

sub procedure name (arguments)

.....

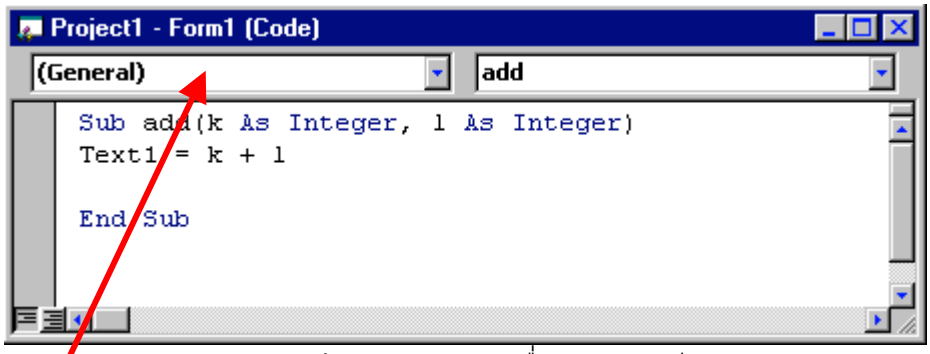
.....

End sub

เมื่อ arguments คือค่าที่โปรแกรมหลักส่งให้กับโปรแกรมย่อย

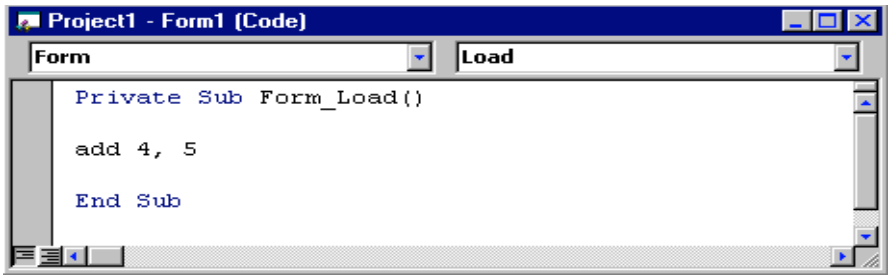
ตัวอย่าง 3.1

สร้างโปรแกรมย่อยสำหรับ บวกเลข

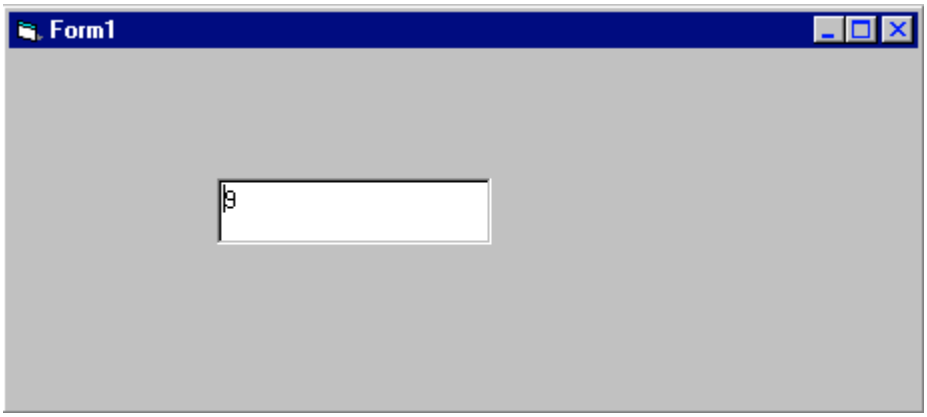


รูป 3.1 แสดงการสร้างโปรแกรมย่อยชื่อ add และมีการรับค่า arguments 2 ค่า

หมายเหตุ โปรดสังเกตุตำแหน่งของการวางโปรแกรมย่อย



รูป3.2 เรียกใช้โปรแกรมย่อย add จาก sub form_load



รูป 3.3 ผลจากการทำงานของโปรแกรมย่อย

3.3. User Defined Function

สำหรับ User Defined Function จะแตกต่างจาก User Defined Procedure ตรงที่ คำขึ้นต้นจะขึ้นด้วย Function และ สามารถส่งค่ากลับให้กับโปรแกรมหรือฟังชั่นที่เรียกใช้ได้ รูปแบบจะเป็นดังนี้

Function function name (arguments) as data type

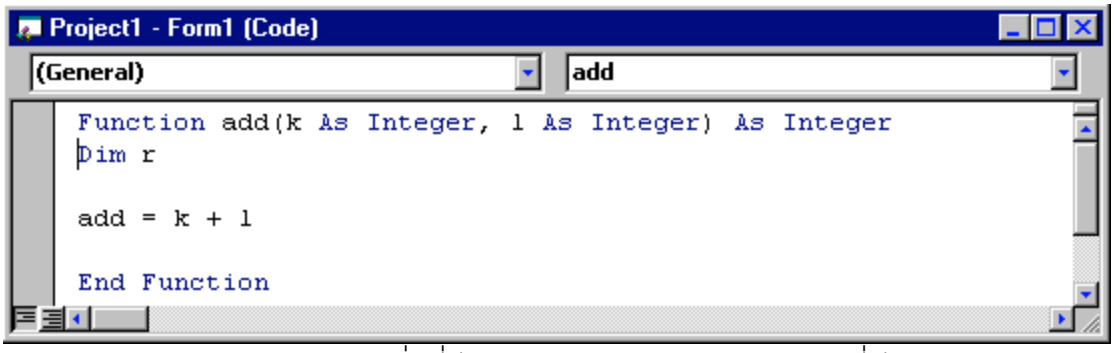
……..

……..

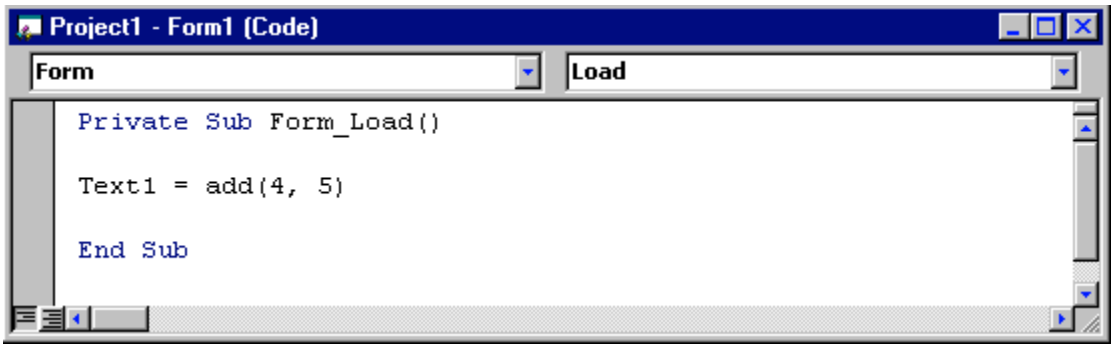
end function

ตัวอย่าง 3.2

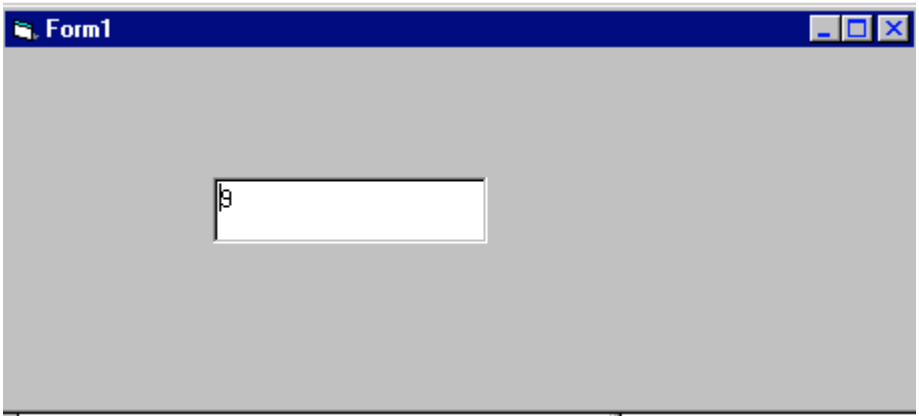
การสร้างฟังชั่นย่อยสำหรับการบวกตัวเลขสองค่า



รูป 3.4 การสร้างฟังชั่นที่มีการรับและส่งค่าระหว่างโปรแกรมที่เรียกใช้



รูป 3.5 เรียกใช้ Function add จาก sub form_load



รูป 3.6 ผลจากโปรแกรมจาก sub form_load

3.3 ประเภทของ Arguments

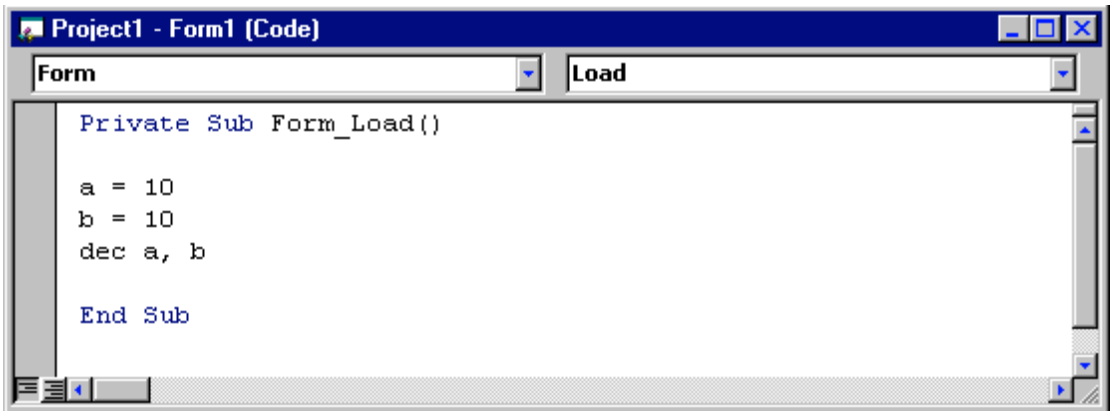
3.3.1 Arguments แบบ By Value arguments แบบนี้เมื่อมีการเปลี่ยแปลงค่าใน function จะไม่สงผลกับตัวแปรของโปรแกรมหรือฟังชั่นที่เรียกใช้

3.3.2 Arguments แบบ Byref ตัวแปรที่ทำหน้าที่เป็น arguments แบบนี้เมื่อมีการเปลี่ยนแปลงค่าในฟังชั่นจะส่งผลต่อตัวแปรต้นฉบับด้วย

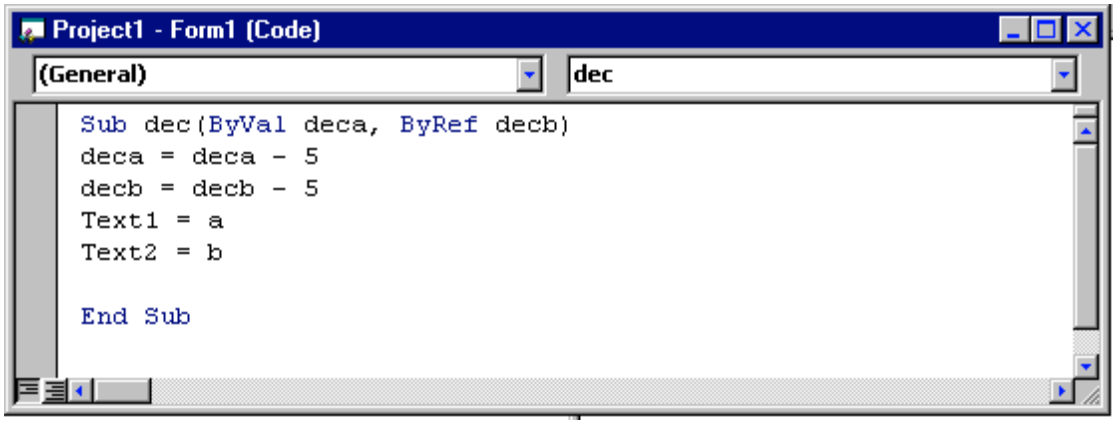
ตัวอย่าง 3.3

แสดงความแตกต่างของ arguments Byval และ Byref

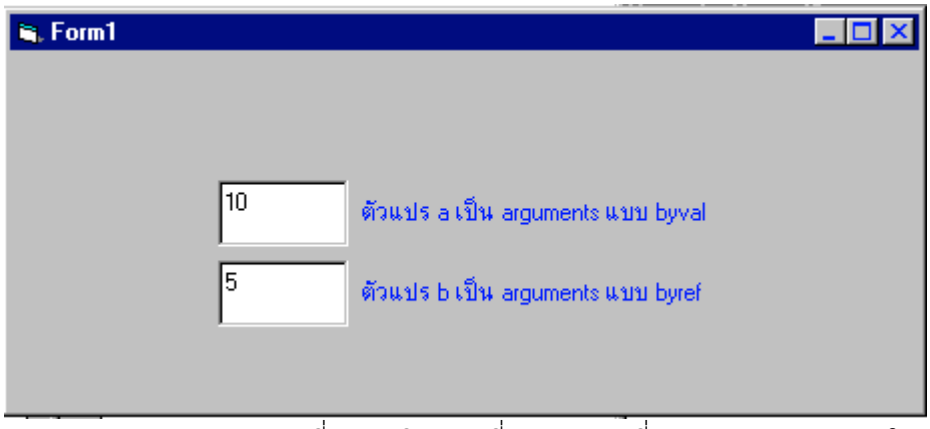
รูปที่ 3.7 ประกาศตัวแปร ที่ general



รูป 3.8 กำหนดค่าให้กับตัวแปร ก่อนเรียกโปรแกรมย่อย dec



รูป 3.9 โปรแกรมย่อยรับค่า arguments 2 แบบ



รูป 3.10 ค่าตัวแปร a มีค่าเท่าเดิม ขณะที่ ตัวแปร b เปลี่ยค่าจากผลของงานจากโปรแกรมย่อย

3.4 ประเภทของข้อมูล

ประเภทของตัวแปร	สัญลักษณ์	พื้นที่การใช้งาน	ขอบเขตการรองรับข้อมูล
string	$	ความของ string+4	ใช้ได้ถึง 65,535 ตัวอักษร
integer	%	2 byte	-32,768 to +32,767
Long	&	4 byte	-2,147,483,648 to +2,147,483,647
single	!	4 byte	-3.402823E+38 to –1.401298E+45 และ +1.401298E-45 to 3.402823E+38
double	#	8 byte	-1.797693134862315D+308 to –4.94066D-324 และ +4.94066D-324 to +1.797693134862315D+308
Currency	@	8 byte	-922337203685477.5808 to +922337203685477.5807
variant	(none)	varies	Any of the other data type

String ใช้เก็บข้อความต่าง ๆ

Integer และ Long ใช้เก็บค่าตัวเลขจำนวนเต็ม

Single และ double ใช้เก็บค่าตัวเลขจำนวนจริง

Currency ใช้เก็บค่าที่เป็นจำนวนเงิน

Variant ใช้เก็บค่าประเภทใดก็ได้ใน 6 ประเภทข้างต้นโดยจะแปรเปลี่ยนตามข้อมูลที่ได้รับ

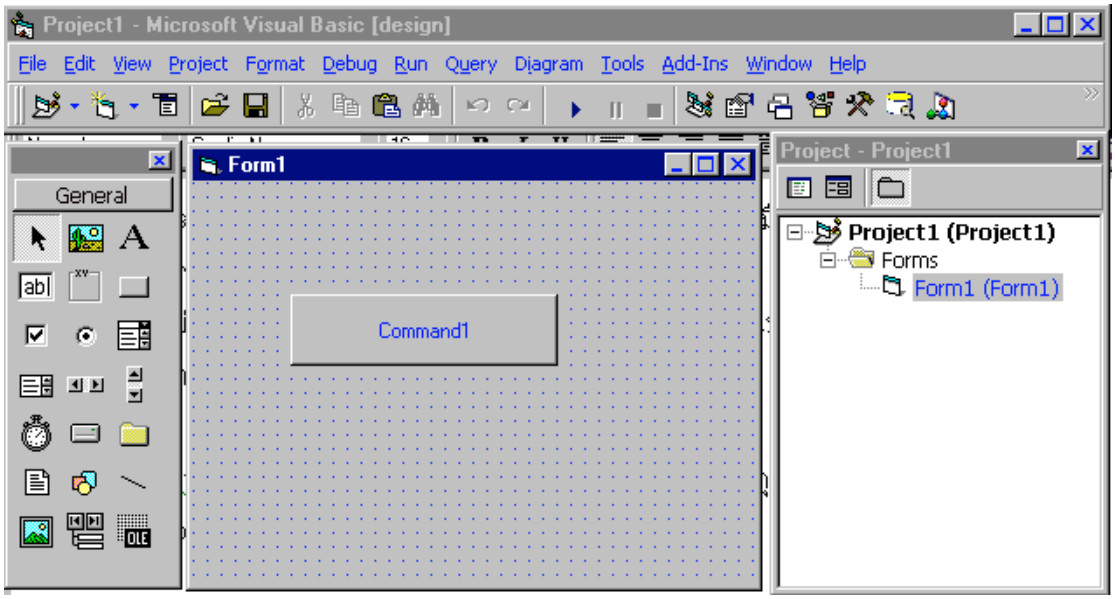
4

ขอบเขตการใช้ งานของ

ตัวแปร , User Defined Procedure และ User Defined Function

ต่อไปเราจะมองระบบงานที่พัฒนาบน visual basic ในเรื่องของการบริหารจัดการข้อมูล การใช้งาน User Defined Procedure และ การใช้ User Defined Function ที่ต้องนำเรื่องนี้มาพูดกันก็เพราะว่าเป็นเรื่องที่ต้องให้ความสำคัญเป็นอย่างมากสำหรับการทำงานของนักพัฒนาโปรแกรมมือใหม่ พูดได้ว่าการวางแผนงานที่ดีก่อนลงมือปฏิบัติงานจริงนั้นเป็นองค์ประกอบที่สำคัญในการทำให้งานนั้น ๆ สำเร็จลุล่วงไปด้วยดี และต่อจากนี้ไปผู้เขียนจะยกตัวอย่างเริ่มตั้งแต่งานที่ประกอบด้วย Form เพียงอย่างเดียวจนไปถึงตัวอย่างที่มี Form,Module มากกว่า 1 Form หรือ 1 Module

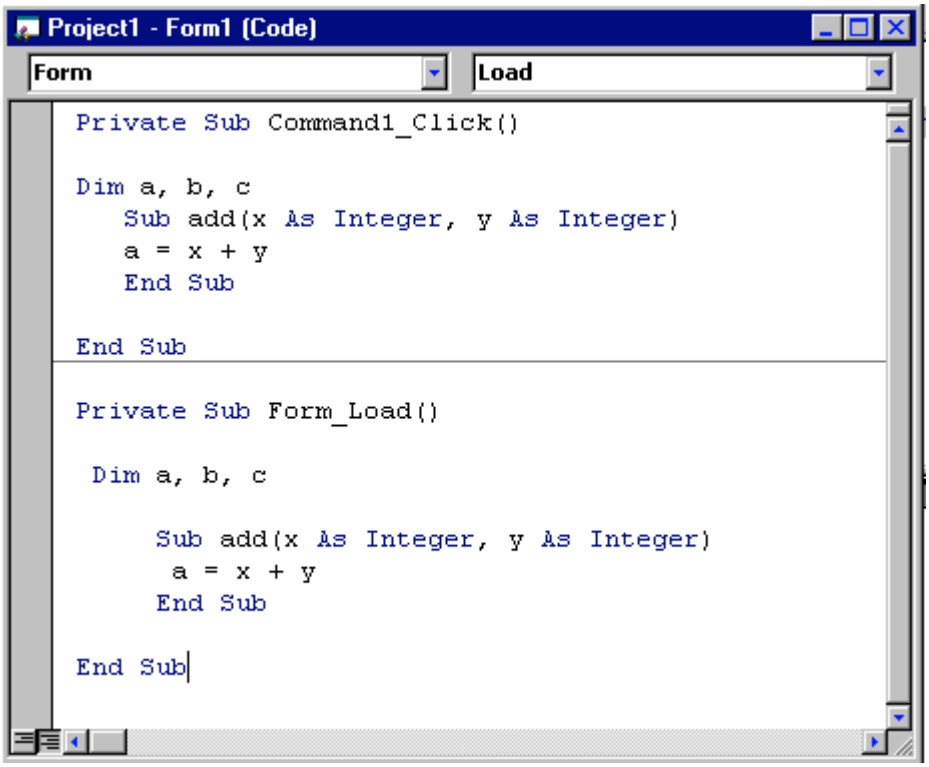
4.1 Project ที่มีองค์ประกอบ Form เพียง 1 Form และมีวัตถุแวดล้อม 2 อย่างคือ form กับปุ่ม command1 ตามรูป 4.1



รูป 4.1 project ขนาด 1 Form

4.1.1 ตัวแปร,โปรแกรมย่อย แบบ private คือตัว แปรที่ถูกจำกัดให้มีการใช้งานหรือมองเห็นได้ในที่ใดที่หนึ่งเท่านั้น

ตัวอย่าง 4.1 ตัวแปร,โปรแกรมย่อย แบบ private

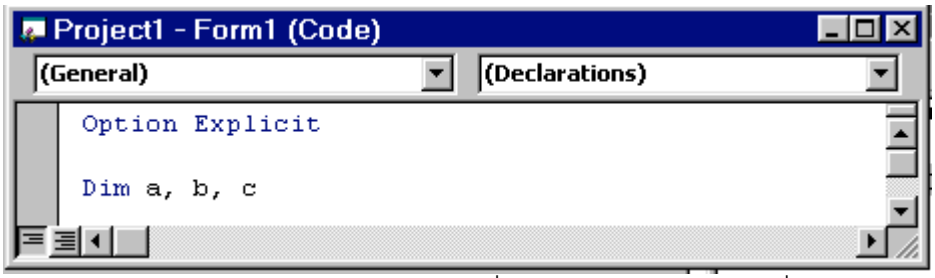


รูป 4.2 ตัวแปร a,b,c และ sub program add() เป็น ตัวแปร และ โปรแกรมย่อยแบบ private

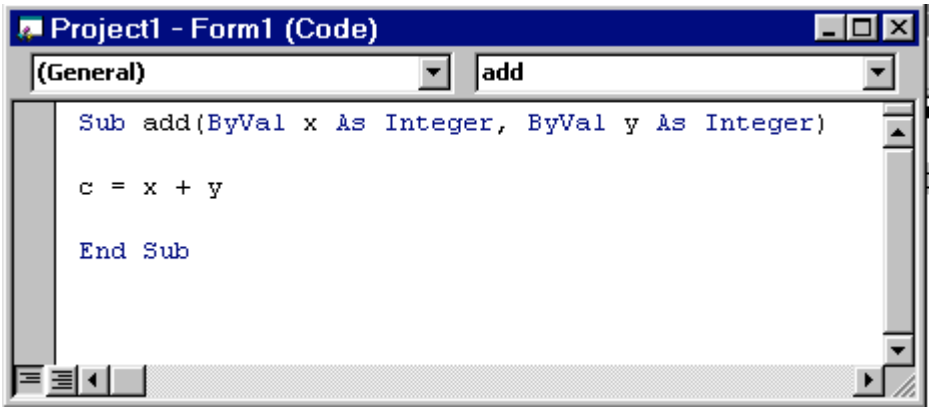
จากรูป 4.2 จะเห็นว่าที่ sub command1_click และ sub form_load มีการประกาศตัวแปร และสร้างโปรแกรมย่อยที่มีชื่อซ้ำกันแต่จะไม่มีความสบสนวุ่นวายเกิดขึ้นในการทำงานแต่อย่างใดเนื่องจากทั้งตัวแปรและโปรแกรมย่อยถูกจำกัดขอบเขตการใช้ภายใน sub program ย่อยเท่านั้น

4.1.2 ตำแหน่งของตัวแปร,Sub program ที่ใช้งานร่วมกันภายใน Form 1 Form

ตำแหน่งของการประกาศตัวแปร และ โปรแกรมย่อย เพื่อให้ทุก ๆ โปรแกรมย่อยที่อยู่ภายใน Form1 สามารถเรียกใช้หรือเข้าถึงข้อมูลได้

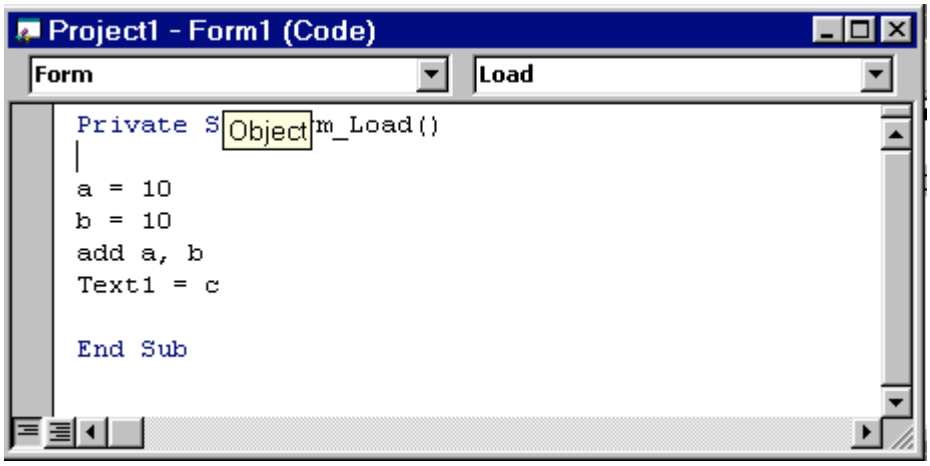


รูป 4.3 ตำแหน่งของการประกาศตัวแปรเพื่อให้ทุก ๆ โปรแกรมย่อยที่อยู่ใน Form1สามารถเข้าถึงได้

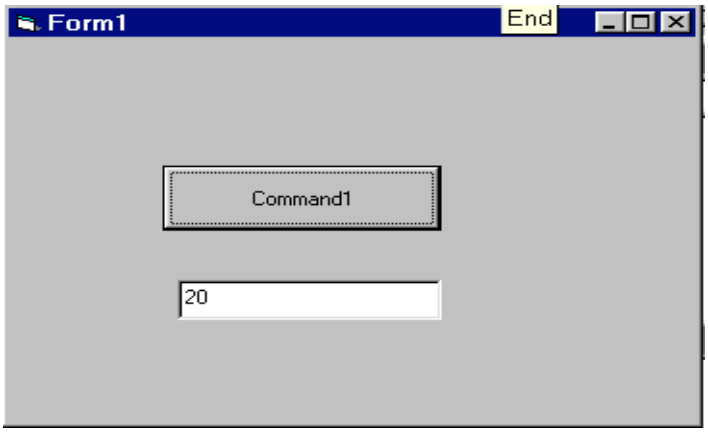


รูป 4.4 ตำแหน่งของ User Defined Procedure เพื่อให้ทุก ๆ โปรแกรมย่อยที่อยู่ใน Form1 สามารถเรียกใช้ได้

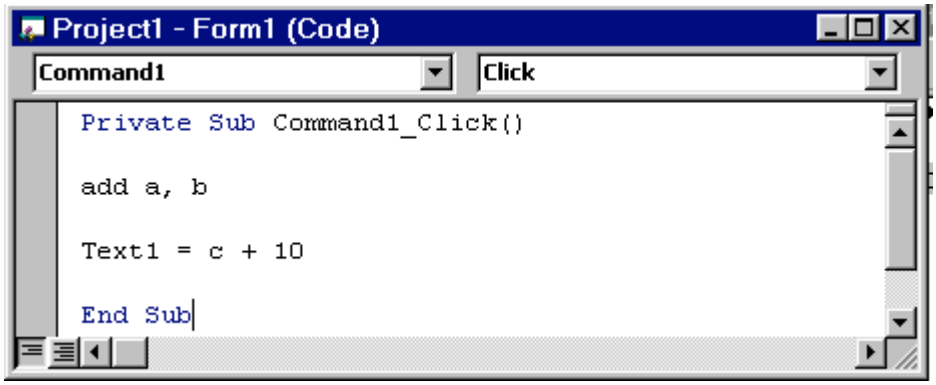
ตัวอย่าง 4.2 แสดงการใช้ ตัวแปรและ User Defined Procedure ที่อยู่ใน general



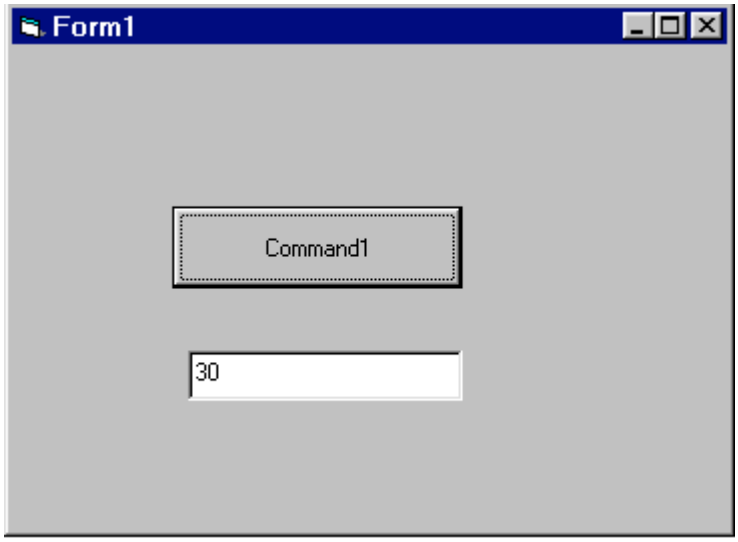
รูป 4.5 แสดงการเรียกใช้โปรแกรมย่อย add จาก sub form_load



รูป4.6 ผลจากการทำงานของโปรแกรมจากรูป 4.5

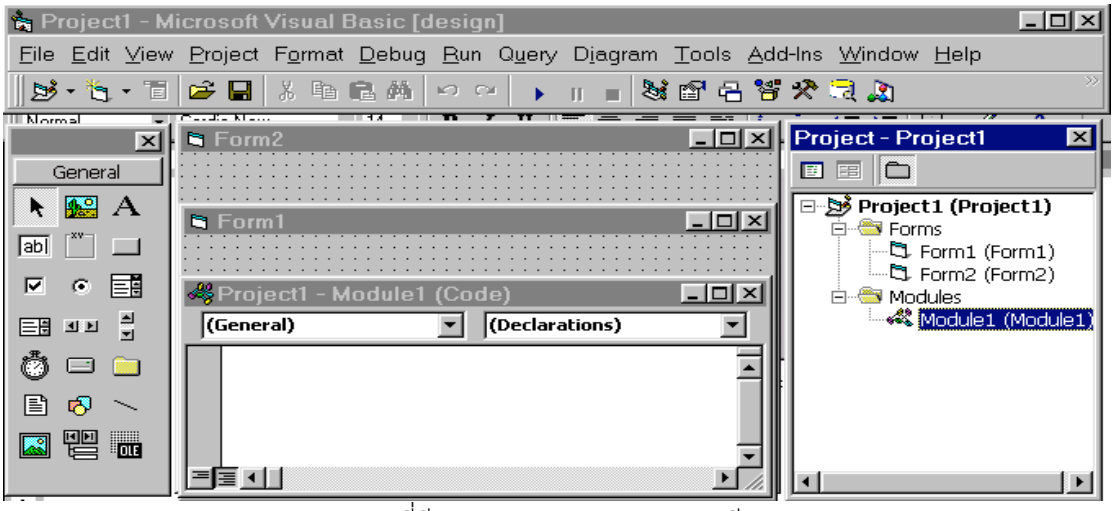


รูป 4.7 การเรียกใช้โปรแกรมย่อย add จาก sub command1_click



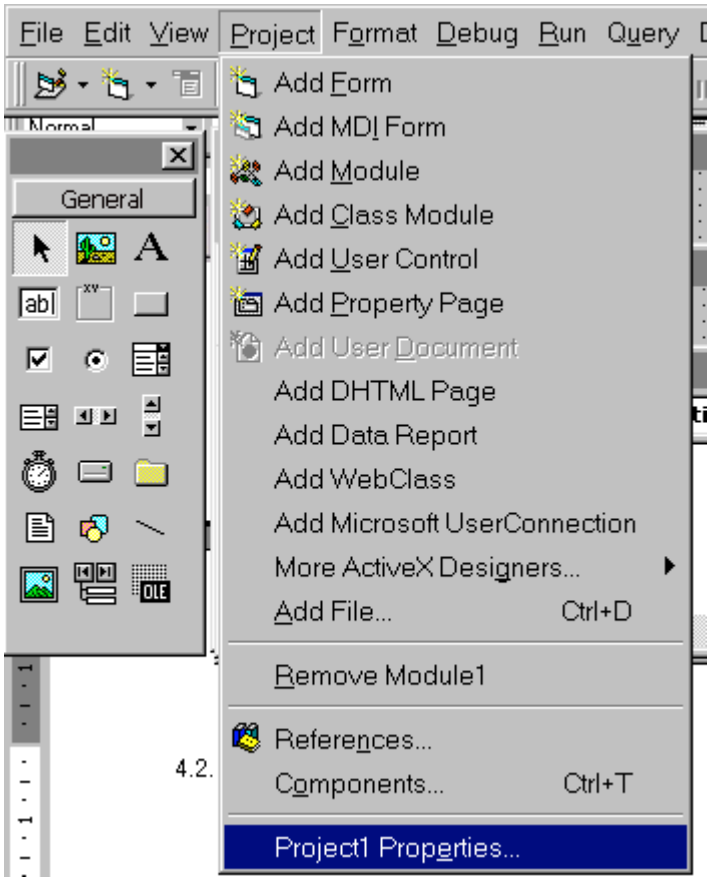
รูป 4.8 ผลที่เกิดขึ้นหลังจาก RUN program แล้ว click command1

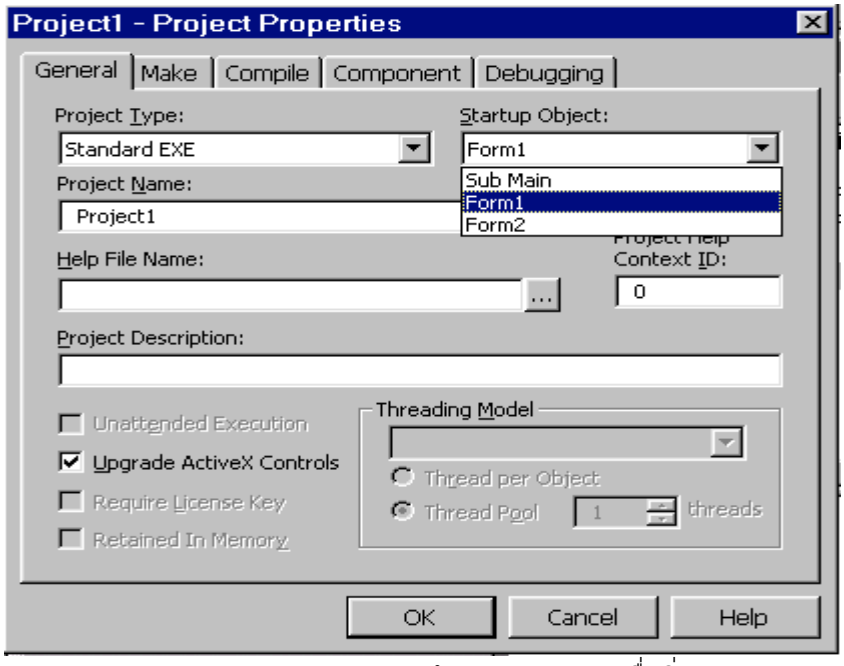
4.2 Project ที่ประกอบด้วย Form มากกว่า 1 Form และ มี Module



รูป 4.9 Project ที่มี Form มากกว่า 1 Form และ มี Module

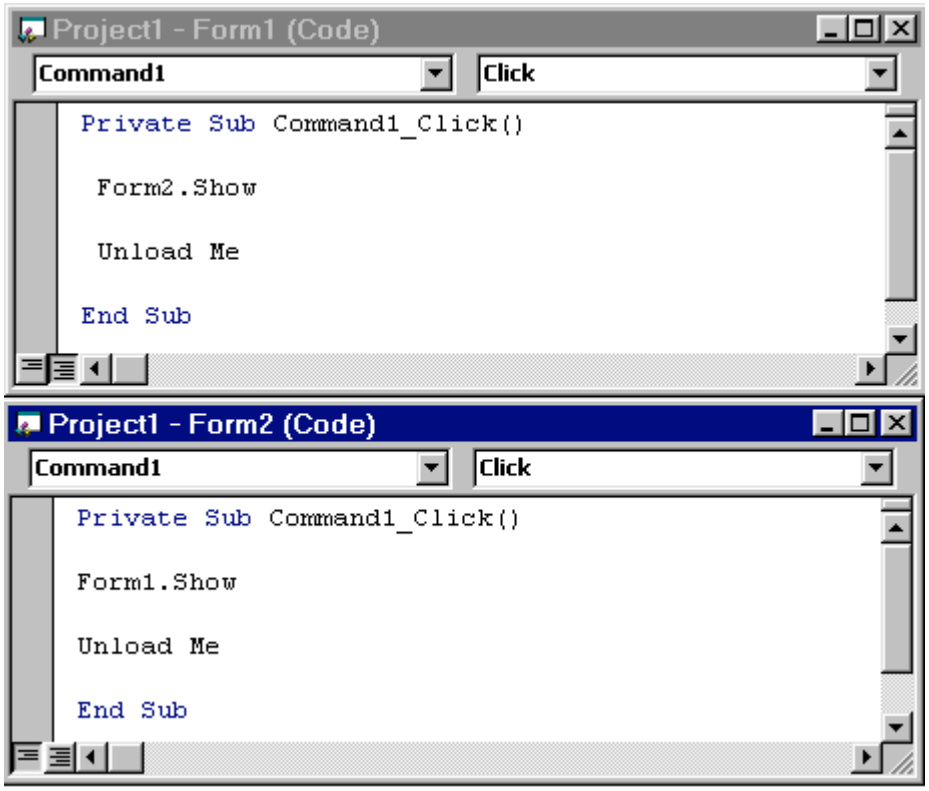
4.2.1 การเลือก start form เมื่อเริ่ม run program





รูป 4.10 การกำหนด start form เมื่อเริ่ม run program

4.2.2 คำสั่งเพื่อใช้ในการเปิดและปิด Form

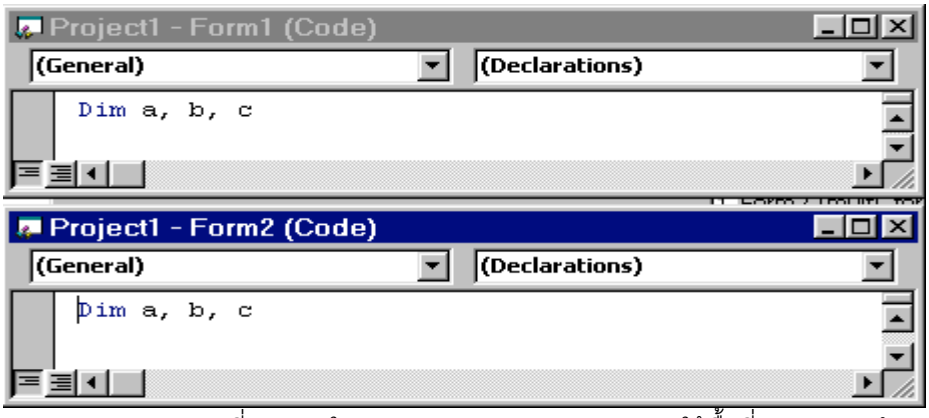


รูป 4.11 พิมพ์คำสั่งที่ sub command1_click ใน form1 และ form2

-ทดลอง run program แล้ว click ปุ่ม command1 ใน form1 ผลจะทำให้ form1 ถูกปิด และ เปิด form2 และเมื่อ click ปุ่ม command1 ใน form2 จะทำให้ form2 ถูกปิด และ จะเปิด form1 ใหม่

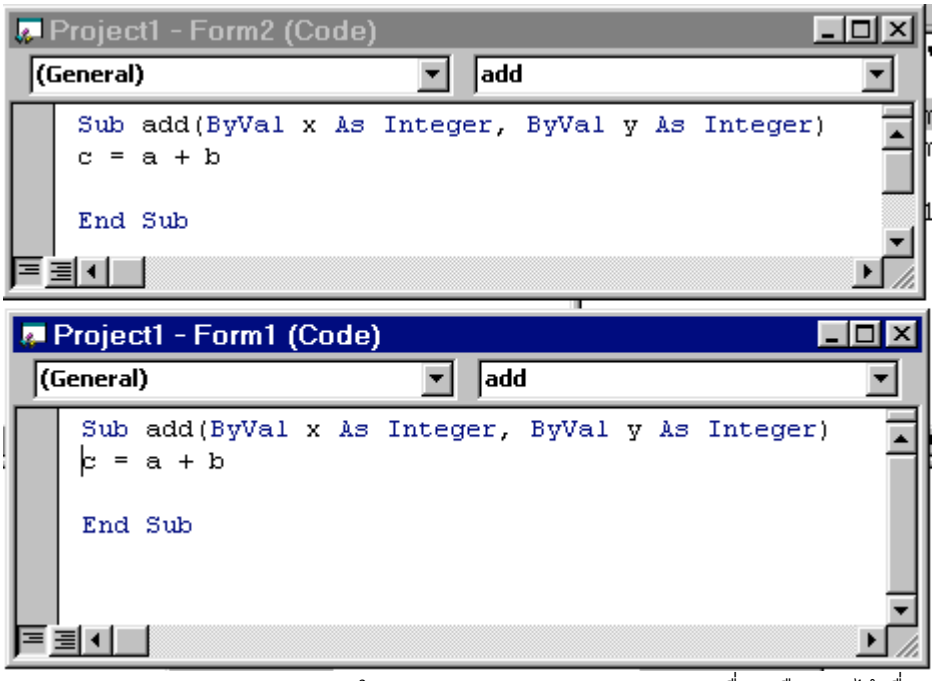
4.2.3 หลักการใช้ตัวแปรเมื่อ project มีมากกว่า 1 form

- ตัวแปรที่ประกาศใน General ของแต่ละ form จะมองเห็นหรือใช้ได้ภายใน form เท่านั้น



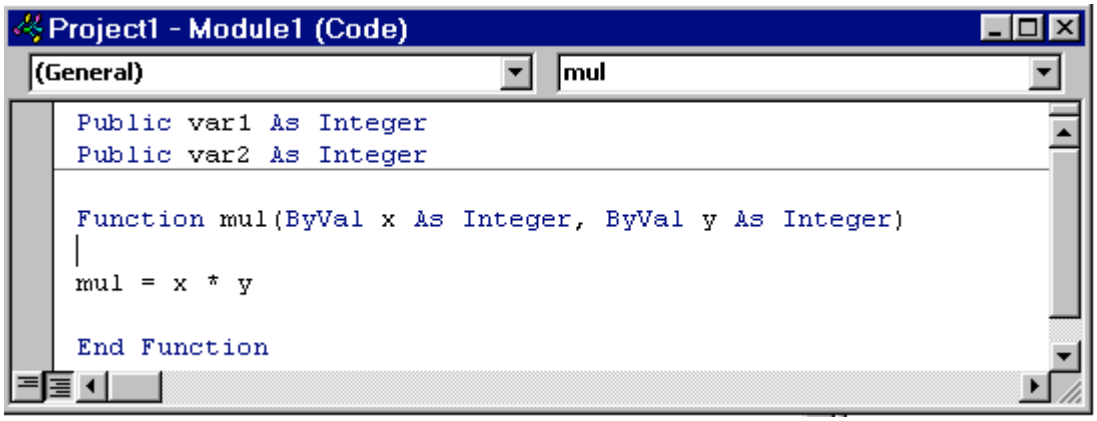
รูป 4.12 ตัวแปร a,b,c ที่ประกาศใน general ของ form1 และ form2ใช้พื้นที่หน่วยความจำแยกจากกัน

- โปรแกรมย่อยที่สร้างใน general ของแต่ละ form จะเรียกใช้ได้จากใน form เท่านั้นไม่สามารถเรียกใช้จากภายนอก form ได้

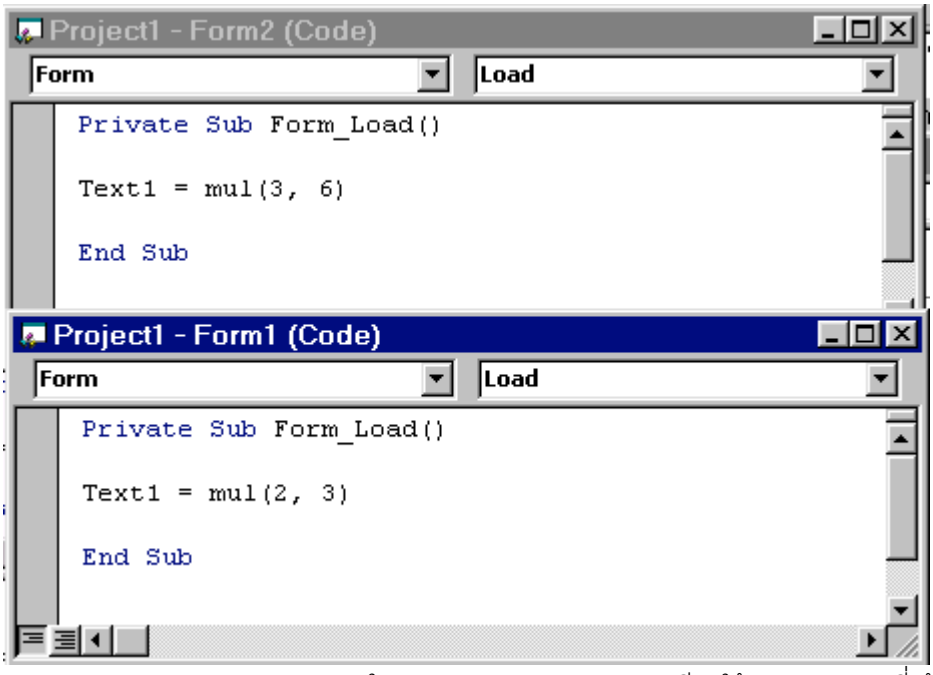


รูป 4.13 sub program add ใน general ของ form1 และ form2 ชื่อเหมือนกันได้เนื่องจากถูกแยกโดย form

4.2.4 พื้นประกาศตัวแปรเพื่อให้แต่ละ form สามารถเข้าถึงได้



รูป 4.14 โปรแกรย่อยจากทุกform สามารถเข้าถึงตัวแปร var1,var2 ได้ เช่นเดียวกัน ทุก form ในproject สามารถเรียกใช้ function mul ได้เช่นกัน



รูป 4.15 sub form_load ใน form1 และ form2 สามารถเรียกใช้ function mul ที่สร้างในModule ได้