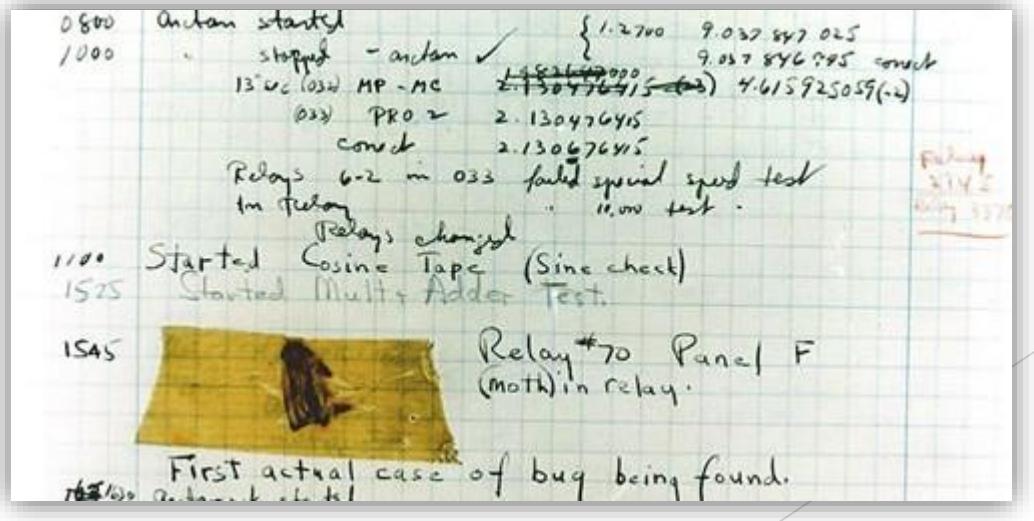
บทที่ 8 การแก้ไขข้อผิดพลาดโปรแกรม (Debugging)

รายวิชา สธ 113 การออกแบบโปรแกรมทางธุรกิจเบื้องต้น

อ.อภิพงศ์ ปิงยศ

What is bug?

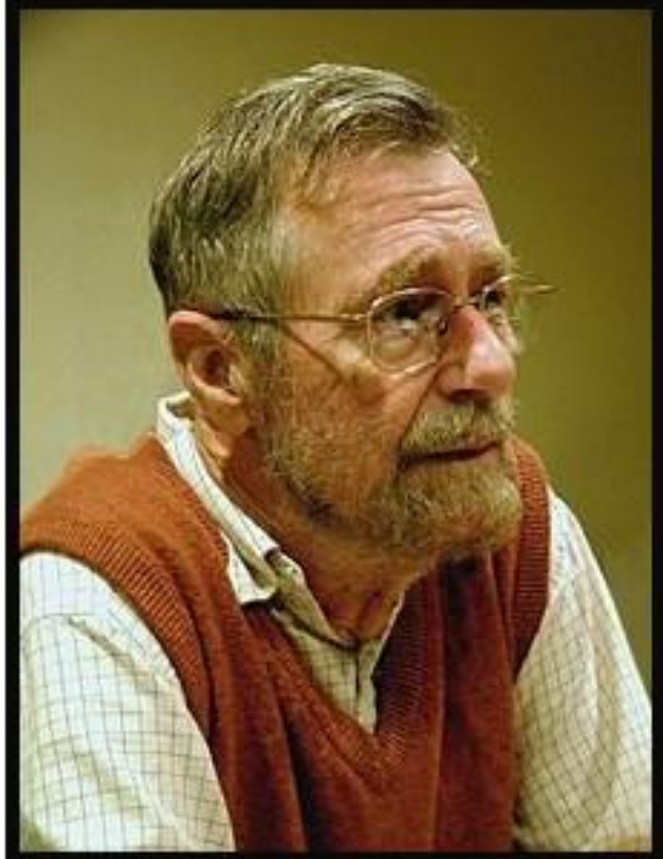
- Bug คือข้อผิดพลาดที่เกิดขึ้นในโปรแกรม
- Bug อาจเกิดจากการเขียนโค้ดผิดรูปแบบ ตรรกะ หรือแม้กระทั่งการใช้งานในโปรแกรมที่ต่างเวอร์ชันกัน
- การแก้ไขข้อผิดพลาดของโปรแกรม เรียกว่า “ดีบัก (Debug)”



BUGS,
BUGS EVERYWHERE

Debugger

- Debugger เป็นเครื่องมือที่ใช้ในการตรวจสอบหาข้อผิดพลาดในโปรแกรม (Bug) ในขณะที่โปรแกรมกำลังรันอยู่ได้
- ปกติแล้วเราจะใช้ค้นหาข้อผิดพลาดทางตรรกะ (Logic Error) มากกว่าการค้นหาข้อผิดพลาดทางไวยากรณ์ (Syntax Error) เพราะข้อผิดพลาดทางไวยากรณ์จะถูกตรวจสอบโดยคอมไพเลอร์อยู่แล้ว
- Debugger สามารถใช้ในการมอนิเตอร์ค่าของตัวแปรทุกตัวที่อยู่ในโปรแกรมได้ ทำให้เราสามารถหาบักที่แท้จริงในโปรแกรมได้
- MS Visual Studio มี Built-in Debugger อยู่แล้ว จึงไม่จำเป็นต้องติดตั้งเพิ่ม



If debugging is the process of removing bugs, then programming must be the process of putting them in.

(Edsger Dijkstra)

izquotes.com

The classic quote.

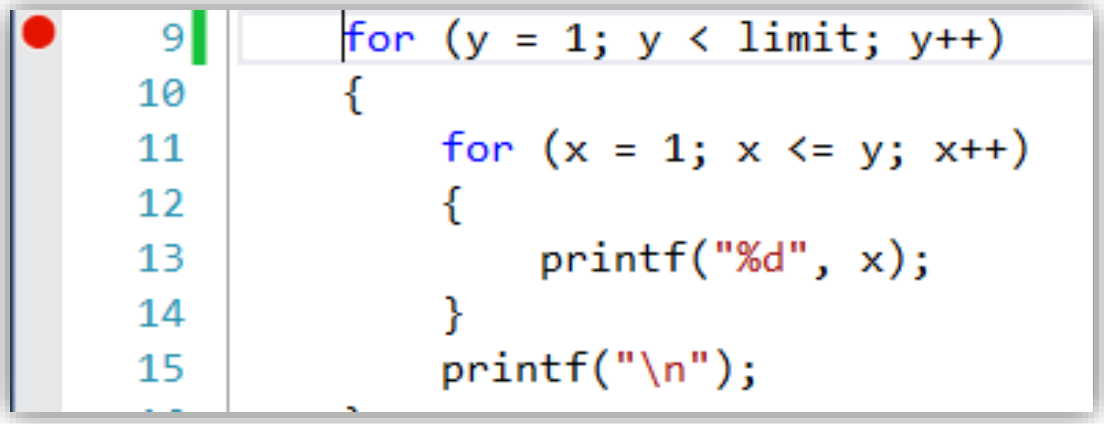
ขั้นตอนการ Debug โปรแกรมขั้นพื้นฐาน

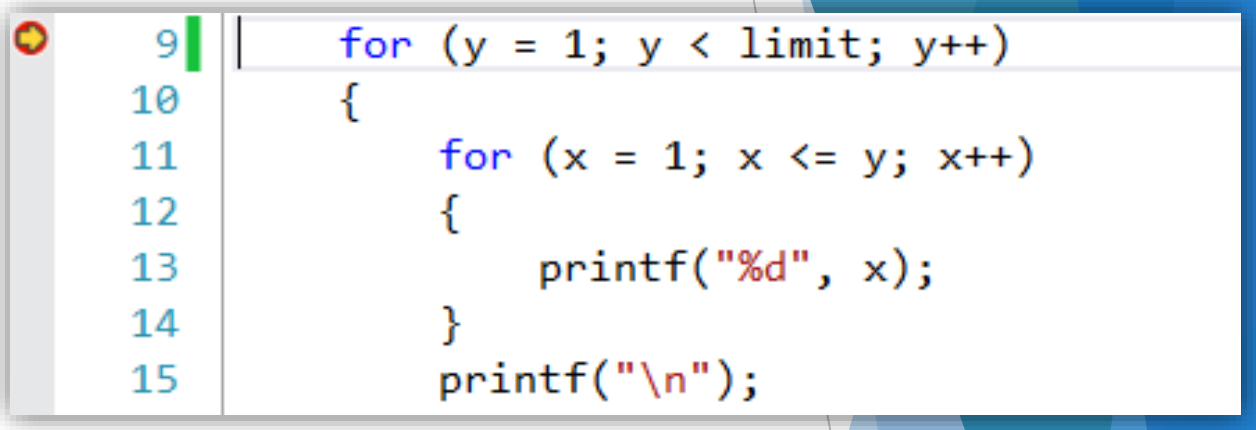
- 1) กำหนด Breakpoint
- 2) Start debugging
- 3) Runtime control

1. กำหนด Breakpoint

- Breakpoint ใช้ในการหยุด Debugger จากการรันโปรแกรมชั่วขณะ และอนุญาตให้โปรแกรมเมอร์ควบคุมการรันต่อไป
- ทำให้โปรแกรมเมอร์สามารถดูค่าตัวแปรและลำดับการทำงานได้ทุก ๆ ขั้นตอนไปจนกว่าโปรแกรมจะรันเสร็จ

การใช้ Breakpoint





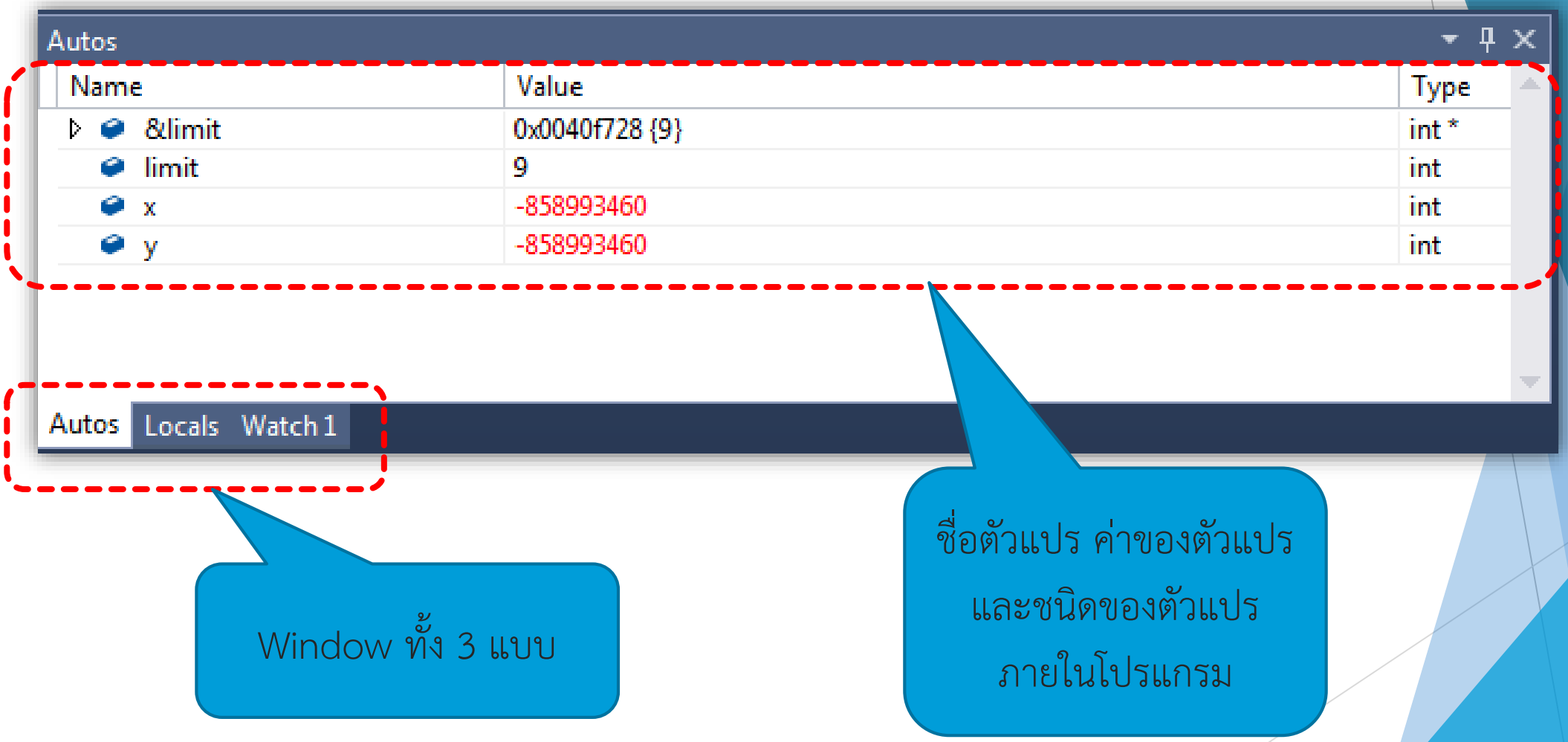
- คลิกซ้ายที่แถบสีเทาหน้าบรรทัดที่ต้องการวาง Breakpoint (สามารถใส่ได้หลายตัว) ปกติจะกำหนดที่คำสั่งรับค่า เงื่อนไข หรือลูป
- หากต้องการนำ Breakpoint ออก ให้คลิกที่ตัว Breakpoint อีกครั้งหนึ่ง

ขณะรันโปรแกรม กระบวนการรันจะมาหยุดที่ Breakpoint เป็นการชั่วคราว

2) Start Debugging

- หลังจากกำหนด Breakpoint เรียบร้อยแล้ว ให้เลือกเมนู Debug
- เลือกคำสั่ง Start Debugging หรือกด F5
- โปรแกรมจะเข้าสู่โหมด Debug แล้วโปรแกรมจะรันคำสั่งต่างๆ จนมาหยุดที่บรรทัดที่มี Breakpoint
- ลูกศรสีเหลืองจะชี้โค้ดในบรรทัดที่กำลังจะดีบักในขั้นตอนต่อไป

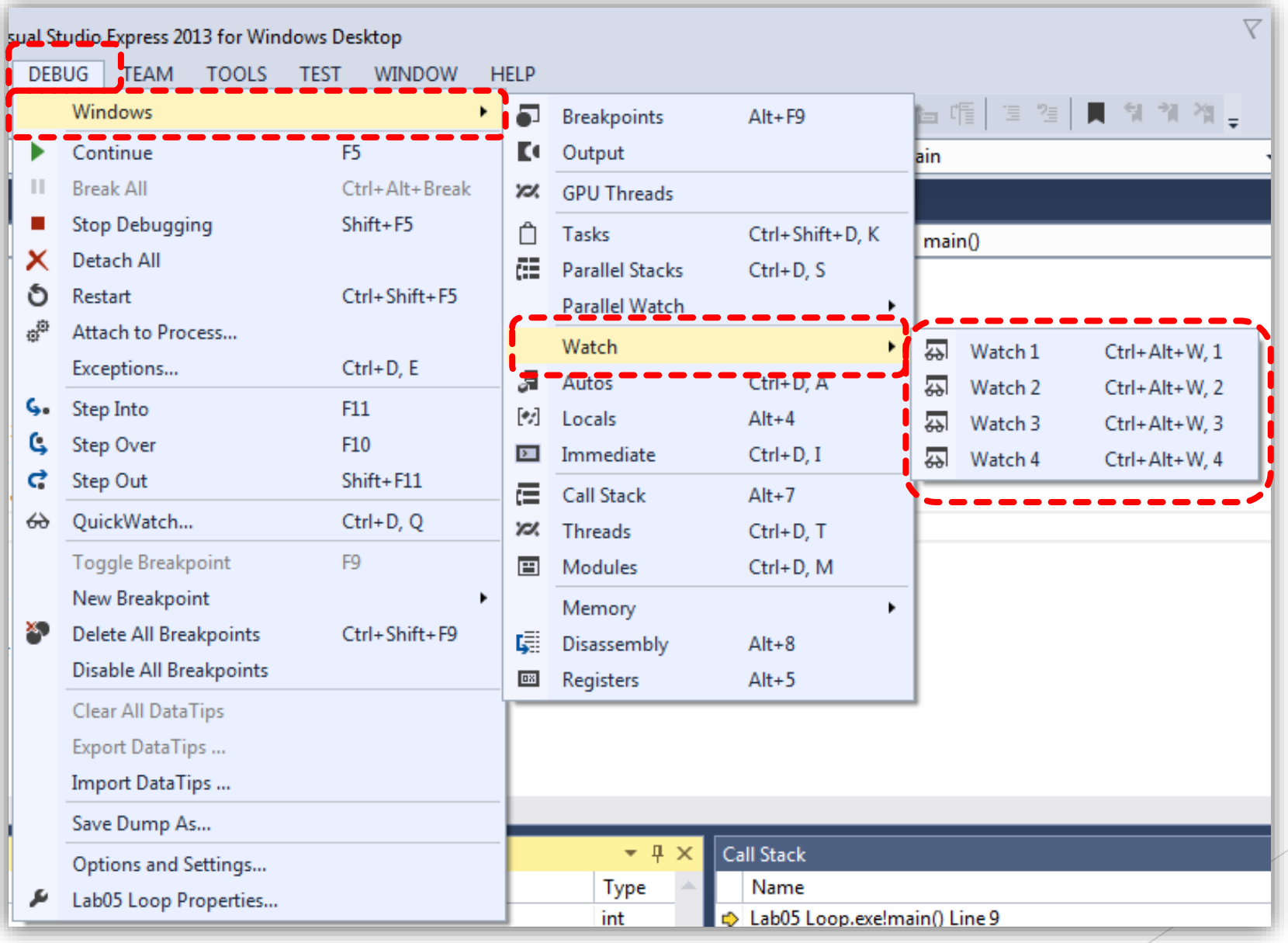
รู้จักกับ Window สำคัญในโหมด Debug



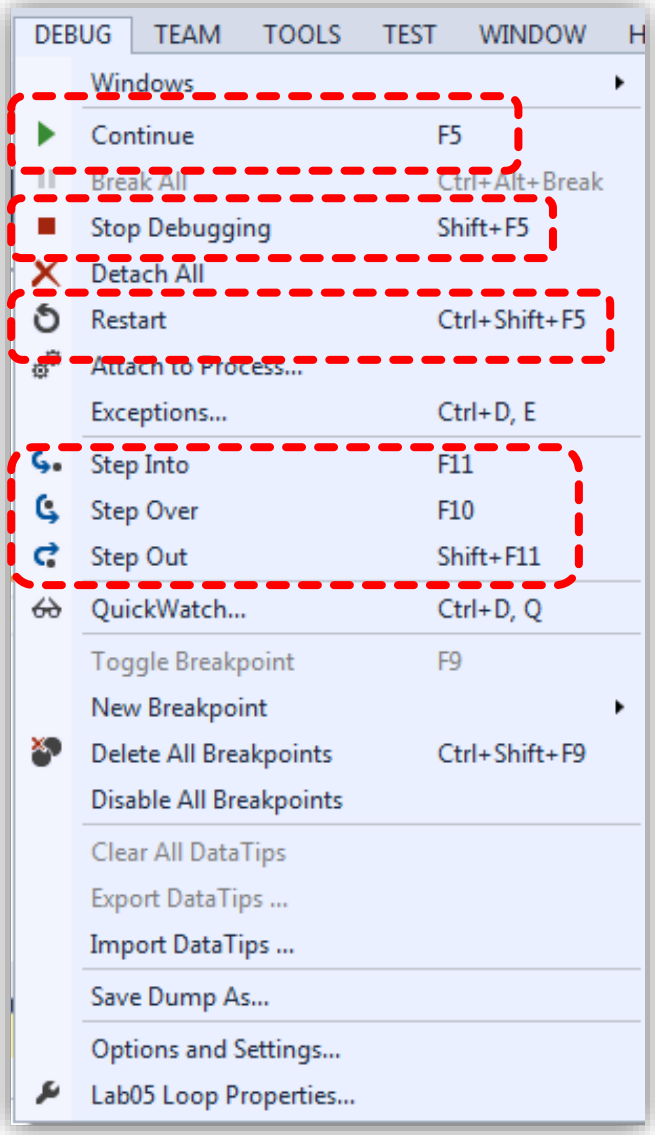
รู้จักกับ Window สำคัญในโหมด Debug [cont.]

- **Autos**: เป็นหน้าต่างที่ใช้แสดงผล*ตัวแปรที่กำลังทำงานอยู่*ในคำสั่งปัจจุบันที่ Debugger กำลังชี้อยู่
- **Locals**: เป็นหน้าต่างที่ใช้แสดงผล*ตัวแปรทั้งหมดที่อยู่ในโปรแกรม*ที่เรากำลังทำการ Debug อยู่ ข้อดีของหน้าต่างนี้คือเราสามารถเปลี่ยนแปลงค่าในตัวแปรในขณะ Debug ได้โดยตรง
- **Watch**: เป็นหน้าต่างที่แสดงผล*ตัวแปรที่โปรแกรมเมอร์สนใจเท่านั้น* วิธีการคือการแดรกเอาชื่อตัวแปรที่สนใจมาใส่ในหน้าต่างนี้ ซึ่งหน้าต่าง Watch นี้สามารถสร้างได้หลายหน้าต่างพร้อม ๆ กัน

การเพิ่มหน้าต่าง Watch



3) Runtime Control

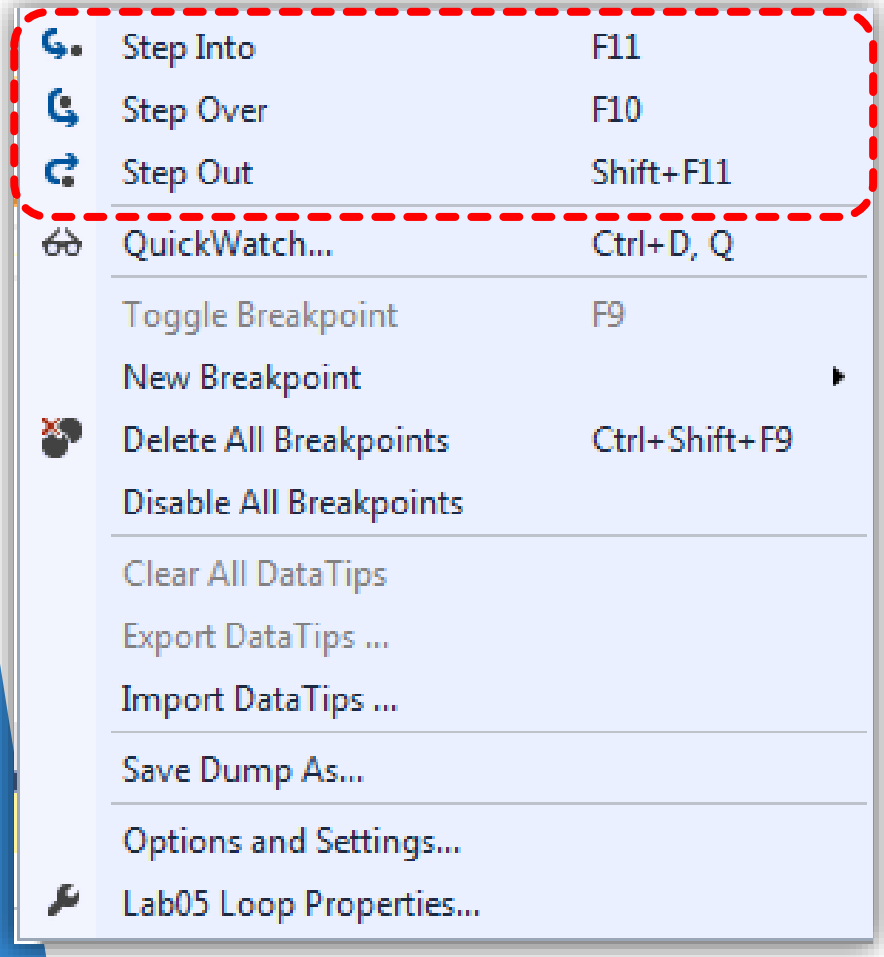


Continue: เป็นการสั่งให้โปรแกรมรันต่อเนื่องไปจนถึง Breakpoint ถัดไป หรือทำงานตามคำสั่งในลูปทั้งหมดโดยรวบรัด

Stop Debugging: เป็นการสั่งให้โปรแกรมยุติการดีบัก

Restart: เริ่มดีบักใหม่ จะใช้ในกรณีที่มีการเปลี่ยนแปลงโค้ดในโปรแกรมขณะดีบัก

3) Runtime Control [cont.]

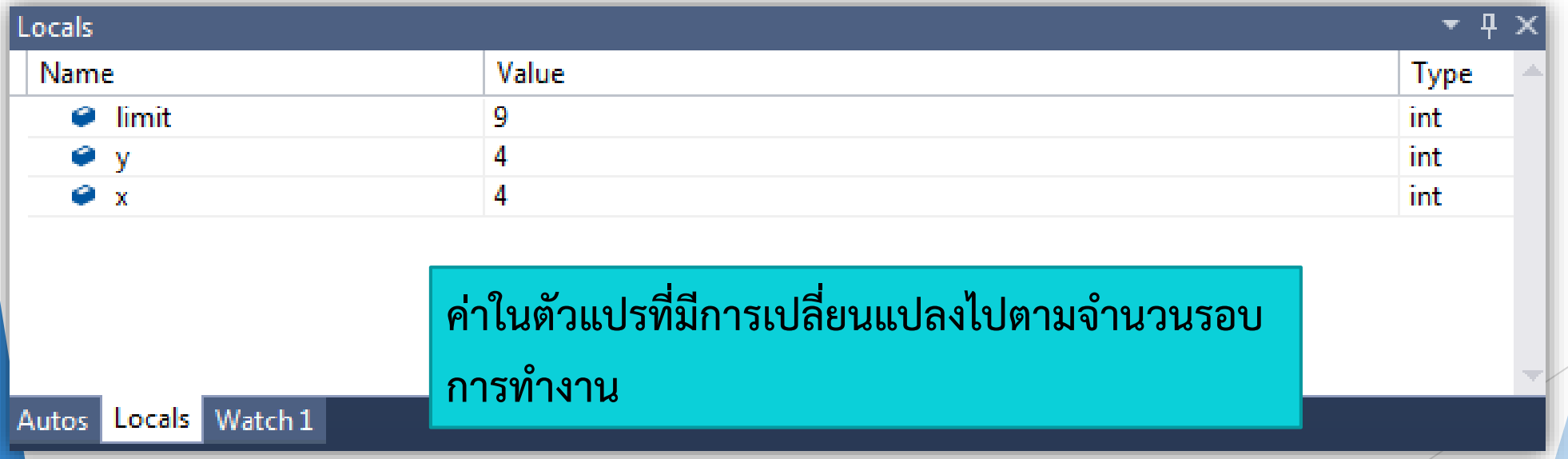


Step Into: เป็นการสั่งให้โปรแกรมดีบักโค้ดในบรรทัดถัดไป *หากบรรทัดนั้นมีการเรียกใช้ฟังก์ชั่นอื่นด้วย ดีบักเกอร์จะ jump ไปดีบักในฟังก์ชั่นนั้นๆ*

Step Over: เป็นการสั่งให้โปรแกรมดีบักโค้ดในบรรทัดถัดไป *แต่ดีบักเกอร์จะไม่ jump ไปดีบักในฟังก์ชั่นที่ถูกเรียกใช้* โดยการดีบักในโปรแกรมปกติที่ไม่มีความซับซ้อนมาก มักจะใช้คำสั่งนี้ในการดีบัก

Step Out: คล้ายกับการทำงานในคำสั่ง Continue

คำสั่ง Runtime Control แบบรวดเร็ว



ค่าในตัวแปรที่มีการเปลี่ยนแปลงไปตามจำนวนรอบการทำงาน

Debug Tips

- ขณะเข้าโหมดดีบัก เราสามารถนำเมาส์ไปชี้ตัวแปรที่เราต้องการดูค่าได้โดยตรง
- เราสามารถเปลี่ยนแปลงค่าที่อยู่ในตัวแปรได้ โดยการเปลี่ยน Value ในหน้าต่าง Locals ด้วยการดับเบิ้ลคลิกค่าที่จะเปลี่ยน กรอกค่าใหม่ลงไป แล้วกด Enter เพียงเท่านี้ค่าในตัวแปรก็จะเปลี่ยนแปลงไป
- เราสามารถสั่งข้ามการทำคำสั่งที่เราไม่ต้องการได้ โดยการเลื่อนลูกศรสีเหลืองไปยังบรรทัดอื่นที่เราต้องการ

ศึกษาเพิ่มเติมได้ที่

- http://pstudiodev.blogspot.com/2012/04/visual-studio-2010.html