

การหาข้อผิดพลาดของโปรแกรม (Bug) ด้วยการ Debug





ประเภทของ Error

- ❖ **Syntax Error** คือ เขียนคำสั่งผิดหลักของโครงสร้างภาษา จนไม่สามารถ Compile ได้ ซึ่งเป็นความผิดพลาดที่ร้ายแรงที่สุด การแก้ไข error ชนิดนี้นักพัฒนาควรอ่านคำเตือนที่ Compiler แสดงออกมา
- ❖ **Runtime Error** คือ ผิดขณะรัน โปรแกรม error นี้จะปรากฏขณะรัน โปรแกรมเท่านั้น เช่น การใช้คำสั่ง scanf โดยที่ไม่มี & หน้าตัวแปรจะแสดงหน้าต่าง error การแก้ไขให้ดูว่า error เกิดขึ้นช่วงใด เช่น ช่วงรับค่าตัวที่สอง ก็ย้อนกลับไปดูที่บรรทัดคำสั่งรับค่าตัวที่สอง
- ❖ **Logical Error** คือ การกำหนดนิพจน์เงื่อนไขผิดพลาด error ประเภทนี้แก้ยากที่สุด เพราะจะไม่มีการฟ้องอะไรออกมา นอกจากผลลัพธ์ที่ไม่ถูกต้องเท่านั้น การแก้ปัญหาที่ดีที่สุด คือ Debug



Bug และ Debug

- ❖ **Bug** คือ ความบกพร่องของโปรแกรมที่ทำให้การทำงานผิดพลาด หรือหยุดชะงัก
- ❖ คำว่า "**Bug**" ที่แปลว่าแมลงนั้น มีที่มาจากคอมพิวเตอร์สมัยก่อนที่ใช้หลอดสุญญากาศมักจะมีแมลงเข้าไป ซึ่งจะทำให้เกิดการลัดวงจร ทำงานไม่ได้ หลังจากนั้นจึงนำคำนี้มาใช้กับโปรแกรมคอมพิวเตอร์ด้วย
- ❖ **Debug** คือ การตรวจสอบแก้ไขจุดบกพร่องของโปรแกรม เพื่อให้โปรแกรมทำงานได้ถูกต้อง
- ❖ **Debugger** คือ โปรแกรมควบคุมการ Debug ซึ่งมักจะมีอยู่แล้วในโปรแกรมที่ใช้ในการเขียนโค้ด (IDE ต่างๆ เช่น CodeBlocks)



ขั้นตอนการ Debug

1. กำหนด Break Point ซึ่งหมายถึง ตำแหน่งบรรทัดที่จะเริ่มต้นตรวจสอบ
 - การกำหนด Break Point ที่บรรทัดใดขึ้นกับการตั้งข้อสงสัยของโปรแกรมเมอร์
2. สั่งให้เริ่ม Debug
 - Debugger จะรันทุกบรรทัดที่อยู่ก่อน Break Point และหยุดที่บรรทัดที่กำหนด Break Point ไว้
3. โปรแกรมเมอร์จะเริ่มสั่งให้ประมวลผลไปที่ละบรรทัดๆ พร้อมกับดูค่าของตัวแปรต่างๆ ว่ามีการเปลี่ยนแปลงไปอย่างไร หากพบแล้วให้หยุดการ Debug และแก้ไขให้ถูกต้อง

ตัวอย่างโปรแกรมที่มี Bug

โปรแกรมคิดส่วนลดราคาสินค้า ถ้าราคามากกว่า 300 บาท ลด 8% นอกเหนือจากนี้ลด 4%

```
#include <stdio.h>
void main() {
    float price, discount;
    printf("Enter price (baht):");
    scanf("%f", &price);

    if (price > 300) {
        discount = 8/100 * price;
    } else {
        discount = 4/100 * price;
    }

    printf("Discount: %.2f Baht\n", discount);
    printf("Total    : %.2f Baht\n", price - discount);
}
```



```
C:\Users\Theerayut\Desktop\La...
Enter price (baht):600
Discount: 0.00 Baht
Total    : 600.00 Baht
```



การกำหนด Break Point

❖ คลิกที่หลังหมายเลขบรรทัด ในที่นี้จะเริ่ม Debug ที่บรรทัดที่ 7

The screenshot shows the Code::Blocks IDE window titled "hello.c [Lab3] - Code::Blocks 10.05". The menu bar includes File, Edit, View, Search, Project, Build, Debug, wxSmith, Tools, Plugins, Settings, and Help. The toolbar contains various icons for file operations, search, and execution. The main editor window shows the following C code:

```
1 #include <stdio.h>
2 main() {
3     float price, discount;
4     printf("Enter price (baht):");
5     scanf("%f", &price);
6
7     if (price > 300)
8         discount = 8/100 * price;
9     else
10        discount = 4/100 * price;
11
12    printf("Discount: %.2f Baht\n", discount);
13    printf("Total    : %.2f Baht\n", price - discount);
14 }
15
```

A red dot on the left margin of line 7 indicates a break point. A callout box with a blue border and a pointer to the red dot contains the text "จุดแสดง Break Point". The status bar at the bottom shows "C:\U\ WINDOWS-874", "Line 15, Column 1", "Insert", "Read/Write", and "default".



ปุ่มคำสั่งในการ Debug



ปุ่ม Next line: สั่งให้
ทำงานบรรทัดถัดไป

ปุ่มหยุด Debug

ปุ่มเริ่มต้น Debug โดย
จะรันไปจนถึงจุด Break Point


ปุ่ม Step into: สั่งให้ทำงานโดยเข้าไปภายในฟังก์ชัน

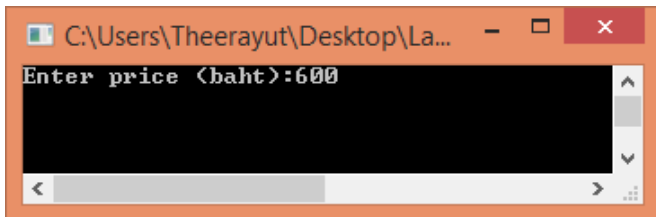
* หมายเหตุ

การ Debug จะทำได้กับ Project เท่านั้น หากไม่ใช่โค้ดที่สร้างใน Project เมนู Debug จะไม่สามารถเลือกได้



เริ่มต้น Debug


- ❖ เริ่ม Debug โดยคลิกที่ปุ่ม  บรรทัดของโค้ดที่อยู่ก่อน Break Point จะเริ่มทำงาน ในที่นี่จะรับค่าทางคีย์บอร์ดก่อน ให้กรอก 600 แล้วกด Enter
- ❖ เมื่อกลับมาที่ CodeBlocks จะมีรูปสามเหลี่ยมสีเหลือง ซึ่งหมายถึงบรรทัดที่จะทำงานต่อไป



```
main.c [rrr] - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main.c x *dsds.c x
1 #include <stdio.h>
2 void main() {
3     float price, discount;
4     printf("Enter price (baht):");
5     scanf("%f", &price);
6
7     if (price > 300) {
8         discount = 8/100 * price;
9     } else {
10        discount = 4/100 * price;
11    }
12
13    printf("Discount: %.2f Baht\n", discount);
14    printf("Total   : %.2f Baht\n", price - discount);
15 }
```




การจับตาดูค่าในตัวแปร


- ❖ การ Debug จะมีหน้าต่าง Watches ที่แสดงรายการตัวแปรและค่าที่บรรจุไว้ทั้งหมด ซึ่งจะมีการ update ค่าให้เห็น เมื่อมีการประมวลผลคำสั่งที่เกี่ยวข้องกับตัวแปรนั้นๆ
- ❖ ระหว่างที่มี Debug เราควรเปิดหน้าต่าง Watches เพื่อจับตาดูค่าต่างๆ ในตัวแปร ณ ขณะที่บรรทัดนั้นทำงานอยู่
- ❖ การเปิดหน้าต่าง Watches ให้เลือกที่ปุ่ม  และเลือกเมนู Watches

แสดงรายการและค่าในตัวแปร

Watches	
Function arguments	
Locals	
price	600
discount	0



การสั่งให้ประมวลผลทีละบรรทัด

- ❖ เริ่มตรวจสอบการทำงานของคำสั่งแต่ละบรรทัด โดยคลิกที่ปุ่ม Next Line 
- ❖ Debugger จะเริ่มทำงานบรรทัดปัจจุบัน และเลื่อนรูปสามเหลี่ยมสีเหลืองไปรอที่บรรทัดถัดไป
- ❖ จากตัวอย่างเมื่อพิจารณาค่าของตัวแปร discount ที่เปลี่ยนแปลงไปในหน้าต่าง Watches จะมีค่า discount เป็น 0 นั้นแสดงว่าบรรทัดที่ 8 จะต้องมีการทำงานที่ผิดพลาดแน่นอน

```
main.c x *dsds.c x
1  #include <stdio.h>
2  void main() {
3      float price, discount;
4      printf("Enter price (baht\n");
5      scanf("%f", &price);
6
7      if (price > 300) {
8          discount = 8/100 * price;
9      } else {
10         discount = 4/100 * price;
11     }
```

หากพิจารณาแล้วจะพบว่า 8/100
จะได้ผลลัพธ์เป็น int ดังนั้นเรา
จึงต้องทำการแปลงให้เป็น float
โดยใส่ (float) หน้า 8/100

Watches	
Function arguments	
Locals	
price	600
discount	0



กิจกรรม

โปรแกรมอธิบายวัยตามค่าอายุที่รับเข้ามา ยังมีข้อผิดพลาดอยู่ ให้ Debug เพื่อหาจุดที่ผิดพลาด

```
#include <stdio.h>
void main() {
    int age;
    printf("Please enter your age ");
    scanf("%d", &age);

    if (age < 75) {
        printf("%d years: pretty young\n", age);
    } else if (age = 75) {
        printf("%d years: old human\n", age);
    } else {
        printf("%d years: really old human\n", age);
    }
}
```



กิจกรรม

ใช้ Debug เพื่อตรวจสอบการทำงานของโปรแกรมต่อไปนี้

```
#include <stdio.h>
void main() {
    int k = 2;
    int p = 1;
    switch(p) {
        case 1:
            k = 3;
        case 2:
            k = 2;
        case 3:
            k = 5;
        default:
            k = 1;
    }
    printf("%d\n", k);
}
```