

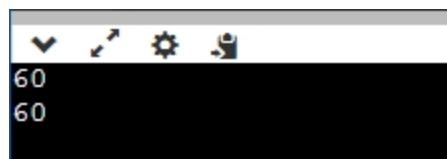
HW9

จงแปลงโปรแกรมภาษา C++ ต่อไปนี้ให้เป็น C#
และเปรียบเทียบผลลัพธ์ กันระหว่าง C++ vs C#
กรณีที่ไม่เหมือนกันให้อธิบายว่าทำมัย?(เหตุผล)

C++

```
main.cpp
1 #include <iostream>
2 using namespace std;
3 //#include <iostream.h>
4
5 int A = 20, B = 5;
6
7 int myfunction(int *p){
8
9     *p = *p + 10;
10    return *p;
11 }
12
13
14 int main( ){
15
16     int result ;
17     A = 20; B = 5;
18     result = A + myfunction(&A);
19     cout << result << endl;
20     A = 20; B = 5;
21     result = myfunction(&A)+A;
22     cout << result << endl;
23     return 0;
24 }
25
26
```

output



ໄປລັງເປັນ C#

```
main.cs
1  using System;
2
3  class Program
4  {
5      static int A = 20, B = 5;
6
7      static int myfunction(ref int p)
8      {
9          p = p + 10;
10         return p;
11     }
12
13     static void Main()
14     {
15         int result;
16
17         A = 20;
18         B = 5;
19         result = A + myfunction(ref A);
20         Console.WriteLine(result);
21
22         A = 20;
23         B = 5;
24         result = myfunction(ref A) + A;
25         Console.WriteLine(result);
26     }
27 }
```

output



เปรียบเทียบผลลัพธ์ กันระหว่าง C++ vs C#

ผลลัพธ์ต่างกันโดยใน C++ ได้ผลลัพธ์เป็น 60 และ 60 ส่วนในภาษา C# จะได้ ผลลัพธ์เป็น 50 และ 60

C++ ได้ผลลัพท์เป็น 60 และ 60 เพราะ ในภาษา C++ นั้นไม่เกิด Function side effect เนื่องจาก ในโปรแกรมตัวอย่างนี้เป็นการ pass แบบ pass by pointer โดยโปรแกรมจะสร้างการอ้างอิง ตัวแปรขึ้นมาโดยใช้ Pointer

ตัวอย่าง

```
main.cpp
1 #include <iostream>
2 using namespace std;
3 //#include <iostream.h>
4
5 int A = 20, B = 5;
6
7 int myfunction(int *p){
8
9     *p = *p + 10;
10    return *p;
11 }
12
13
14 int main( ){
15
16     int result ;
17     A = 20; B = 5;
18     result = A + myfunction(&A);
19     cout << result << endl;
20     A = 20; B = 5;
21     result = myfunction(&A)+A;
22     cout << result << endl;
23     return 0;
24
25 }
```

ในบรรทัดที่ 17 มีการ assign ค่า A เป็น 20 และบรรทัดที่ 18 มีการเรียกใช้ myfunction โดยที่ myfunction เป็นการส่งค่าแบบ pass by pointer ทำให้ในบรรทัดที่ 9 มีการเพิ่มค่า p ขึ้น 10 แต่เป็นการเพิ่มจากค่าของตัวแปรที่ถูกส่งมาไม่ได้เพิ่มที่ค่า A โดยตรงทำให้ไม่เกิด Side effect

C# ได้ผลลัพธ์เป็น 50 และ 60 เพราะ ในภาษา C# เกิด Side effect ขึ้นเนื่องจากโปรแกรมตัวอย่างนี้ใช้การส่งค่าแบบ pass by reference ซึ่งเป็นการส่งตัวแหน่งของค่าตัวแปรไปโดยตรง ทำให้เมื่อเกิดการเปลี่ยนแปลงค่าทำให้ค่าของตัวแปรนั้นเปลี่ยนไปด้วย

ตัวอย่าง

```
main.cs
1  using System;
2
3  class Program
4  {
5      static int A = 20, B = 5;
6
7      static int myfunction(ref int p)
8      {
9          p = p + 10;
10         return p;
11     }
12
13     static void Main()
14     {
15         int result;
16
17         A = 20;
18         B = 5;
19         result = A + myfunction(ref A);
20         Console.WriteLine(result);
21
22         A = 20;
23         B = 5;
24         result = myfunction(ref A) + A;
25         Console.WriteLine(result);
26     }
27 }
```

ในบรรทัดที่ 17 มีการ assign ค่า A เป็น 20 และบรรทัดที่ 19 มีการเรียกใช้ myfunction โดยที่ myfunction เป็นการส่งค่าแบบ pass by reference โดยในบรรทัดที่ 9 มีการเพิ่มค่า p ขึ้น 10

โดยที่ค่า p นั้นอ้างอิงถึงตำแหน่งของค่า A ที่ถูกส่งเข้าไปทำให้เมื่อมีการเปลี่ยนแปลงค่าจึงถูกเปลี่ยนแปลงที่ค่าจริงของตัวแปร A