Lesson 50 Reference

فى لغة c++ لدينا Reference وهى تشبه pointer فى لغة c

كان هناك مثال على pointer دعنا نتذكر سوياً :

```
#include <iostream>
using namespace std;

void swap(int x, int y) {
  int t;
  t = x;
  x = y;
  y = t;
}
// قمنا بعمل function لتبديل القيم

int main() {
  int x = 5, y = 9;
  cout << x << "\t" << y<<endl;
  // الطباعة قبل التبديل
  swap(x, y);
  //call by value هنا يسمى ب
  cout << x << "\t" << y << endl;
  // الطباعة بعد التبديل
}
```

output:

5 9

5 9

لم يتم التبديل، فى لغة c كنا نقوم باستخدام pointer لحل هذه المشكلة:

```
#include <iostream>
using namespace std;

void swap(int *x, int *y) {
  int t;
  t =*x;
  *x = *y;
  *y = t;
}
// قمنا بعمل function لتبديل القيم

int main() {
  int x = 5, y = 9;
  cout << x << "  " << y<<endl;
  // الطباعة قبل التبديل
  cout << "--------\n";
  swap(&x, &y);
  //call by Reference هنا يسمى ب
  cout << x << "  " << y << endl;
  // الطباعة بعد التبديل
}
```

output:

```
5  9
--------
9  5
```

وهنا تم التبديل بعد استخدام & و pointer فى لغة c

لكن فى لغة c++ يمكننا تبديل بواسطة & (Reference)

ويكون كالتالى:

```
#include <iostream>
using namespace std;

void swap(int &x, int &y) {
  //Reference نستخدم
  int t;
  t =x;
  x = y;
  y = t;
}

int main() {
  int x = 5, y = 9;
  cout << x << "  " << y<<endl;
  cout << "--------\n";
  swap(x, y);
  cout << x << "  " << y << endl;
}
```

output:

```
5  9
--------
9  5
```

لكن ما هي الخصائص الموجودة فى pointer , Reference:

```
void* p = &x;
  //void ميزة
  //أننا نستطيع من خلال عمل pointer يشير إلى إى شئ سواء
  //int, float, char
  void &r = x;
  // سيظهر خطأ هنا
  // & لأننا بحاجة إلى معرفة ما يشير إليه
```

مثال على *,& :

```
#include <iostream>
using namespace std;


int main() {
  int x = 5;
  int* p1 = &x;
  int** p2 = &p1;
  int*** p3 = &p2;
  int**** p4 = &p3;
  cout << (p4) << endl;
  //p3 الخاص ب address تقوم بطباعة
  cout << (*p4) << endl;
  //p3 الموجود داخل address تقوم بطباعة
  cout << (**p4) << endl;
  //p2 الموجود داخل address تقوم بطباعة
  cout << (***p4) << endl;
  //p1 الموجود داخل address تقوم بطباعة
  cout << (****p4) << endl;
```

```
  //x الموجود داخل address تقوم بطباعة
}
output:
004FFE60
004FFE6C
004FFE78
004FFE84
5
```

مثال أخر:

```
#include <iostream>
using namespace std;


int main() {
  int x = 5;
  int* p1 = &x;
  int** p2 = &p1;
  int*** p3 = &p2;
  int**** p4 = &p3;
  cout << (p4) << endl;
  x = 9;
  cout << (p4) << endl;
}
output:
0093FB94
0093FB94
```

pointer يستطيع أن يشير إلى أكثر من pointer آخر مع اختلاف القيم
على عكس Reference إذا تغيرت قيمة تغير الباقى
مثال:

```
#include<iostream>
using namespace std;
int main(){
   int a = 50;
   int *p = &a;
   int **p2 = &p;
   int ***p3 = &p2;
   int ****p4 = &p3;
   cout<<"----------------------"<<endl;
   cout<<"Pointers 1: "<<endl;
   cout<< (p4) <<endl;
   cout<< (*p4) <<endl;
   cout<< (**p4) <<endl;
   cout<< (***p4) <<endl;
   cout<< (****p4) <<endl;
   a = 90;
   cout<<"----------------------"<<endl;
   cout<<"Pointers 2: "<<endl;
   cout<< (p4) <<endl;
   cout<< (*p4) <<endl;
   cout<< (**p4) <<endl;
   cout<< (***p4) <<endl;
   cout<< (****p4) <<endl;
   a = 50;
```

```
    int &r = a;
    int &r2 = r;
    int &r3 = r2;
    int &r4 = r3;
    cout<<"--------------------"<<endl;
    cout<<"Reference 1: "<<endl;
    cout<< r <<" "<< r2 <<" "<<r3<<" "<< r4<<endl;
    a = 90;
    cout<<"--------------------"<<endl;
    cout<<"Reference 2: "<<endl;
    cout<< r <<" "<< r2 <<" "<<r3<<" "<< r4<<endl;
}
output:
--------------------
Pointers 1:
0073FD94
0073FDA0
0073FDAC
0073FDB8
50
--------------------
Pointers 2:
0073FD94
0073FDA0
0073FDAC
0073FDB8
90
--------------------
```

Reference 1:

50 50 50 50

Reference 2:

90 90 90 90