References

Chapter 6.13 – Using Reference Variables as Parameters

Starting Out With C++. From Control Structures through Objects (Eighth Edition)

A **reference** is an **alias** for another variable whereas a **pointer** is a **variable** which holds the memory address. A reference is just an alias of another variable. Since it is not a variable, its value cannot be reassigned. The following two cases will show the differences between point and reference.

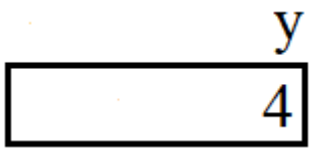
Case 1:

```
int y = 4;
int * yPtr = &y;   // yPtr is a pointer pointing to y
cout << yPtr;      // Display the address of y
```

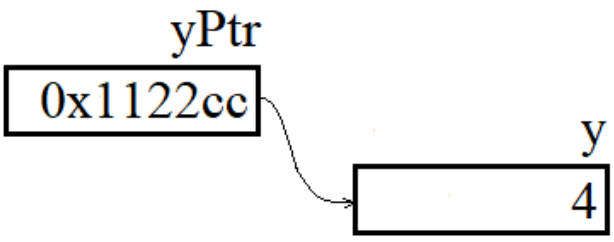
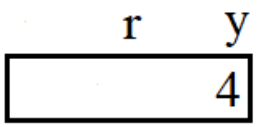
Case 2:

```
int y = 4;
int &r = y;  // r is an alias of y
cout << &r;  // Display the address of y
```

The third statement of each case prints the address of y. But internally they are very different between pointer and reference. So how do they work internally? After the first statement is executed, we have:



Then there are two cases:

	Case 1 - Pointer	Case 2 - Reference
What is it?	A **variable** which holds a memory address.	An **alias** for another variable
Example	int y = 4; int * yPtr = &y;	int y = 4; int &r = y;
What happens internally		
Comment	yPtr is a pointer variable which hold the address of the int y.	The symbol r is an alias of y.

So now you can see that both yPtr and &r display the address of y.

References are generally used as function parameters so that the passed object is not the copy but the object itself.

Note: There are two types of Call-by-References. The first is Call-by-Reference using pointer and the second Call-by-Reference using reference. If we compare the table which listed these two types of Call-by-References in Call-by-Value and Call-by-Reference, we can see that they are exactly the two cases listed above.

Exercises:

- Sample Pointer
- Sample Reference

See Also:

Call-by-Value and Call-by-Reference