Pointer (1A)

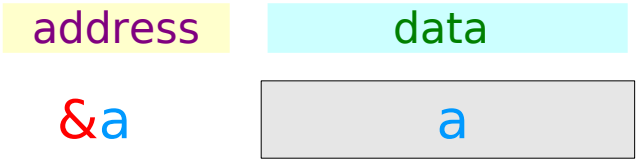
Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice.

Variable

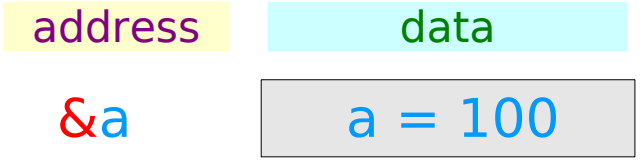
```
int    a;
```

a can hold an *integer*




```
a = 100;
```

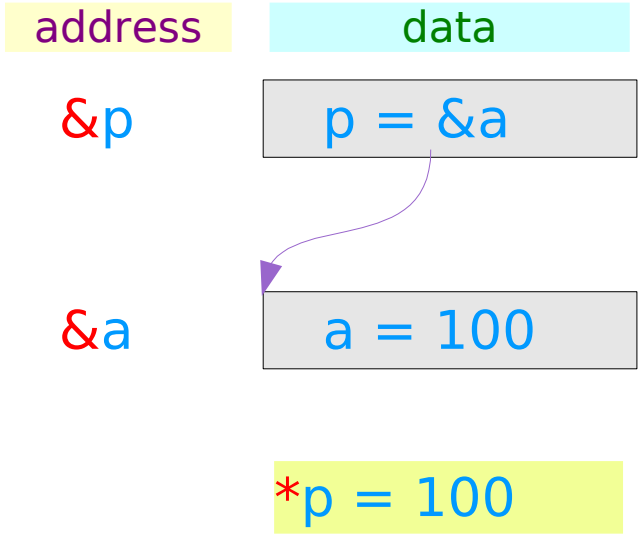
a holds an *integer*



* and & Operator

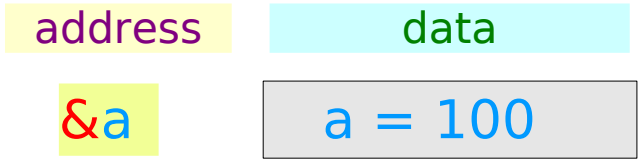
* address

means the value
that is stored at the address



& variable

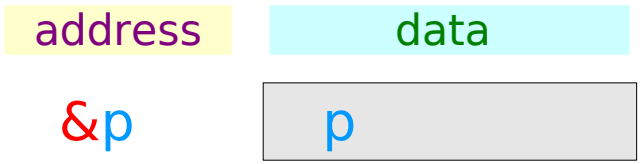
returns the address of a
location where the
variable's value is stored



Pointer (1)

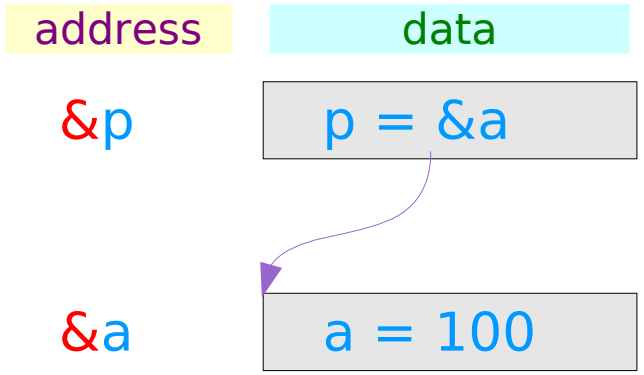
```
int *     p;
```

p can hold an *address*




```
p = &a;
```

p holds the *address* of a



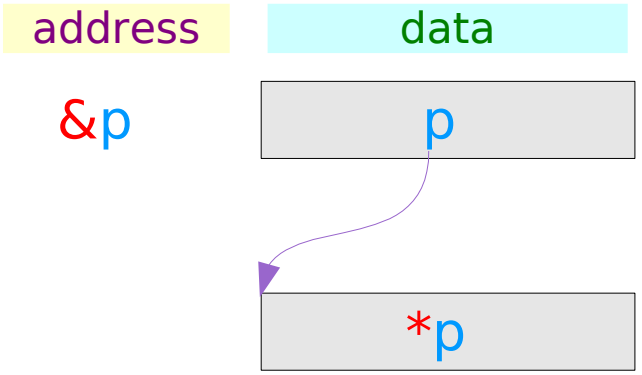
Pointer (2)

```
int *   p;
```

p can hold an *address*

```
int     *p;
```

*p can hold an *integer*



Variable Assignment

```
int     a = 100 ;

int     b =  a ;
```

p can hold an *address*

p is initialized with
the *address* of
integer variable a

address	data
&a	a = 100
&b	b = 100

a and b have the same
integer value

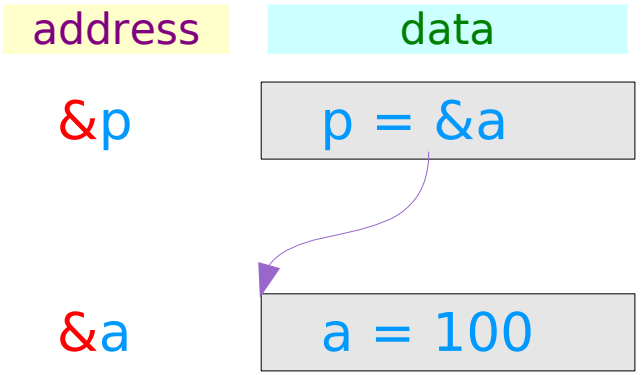
Address Assignment - Pointer

```
int     a = 100 ;

int *   p = & a ;
```

p can hold an *address*

p is initialized with the *address* of integer variable a



a and *p have the same integer value, since &a and p have the same address

Address Assignment - Reference in C++

```
int     a = 100 ;

int &   b = a ;
```

b's address is initialized with a's address

b acts like an integer variable

b holds an *integer*

address	data
&a = &b	a = b

variable b is an ***alias*** of a

a and b have the same integer value, since &a and &b have the same address

References

[1] Essential C, Nick Parlante
[2] Efficient C Programming, Mark A. Weiss
[3] C A Reference Manual, Samuel P. Harbison & Guy L. Steele Jr.
[4] C Language Express, I. K. Chun