C++ Variables

* Regular Variables

```
int   a = 10
float b = 5.2
```

Label	Memory
a	10
b	5.2

* Reference Variables

The reference variable is another name of a regular variable.

```
int a = 10;
int &b = a;
```

a, b | 10

- Variables a and b share the same memory space.
- A change made to variable b also affects the value of variable a.

```
b = 20;
cout << a << " " << b;      // 20 20
```

a, b | ~~10~~ 20

- Reference variables are main used in function parameter passing: pass-by-value vs pass-by-reference

```
int main()                  by-reference
{  int a = 10;              b, a | ~~10~~ 20
   f(a);
}
void f(int &b)              int &b = a
{
    b = 20;
}
```

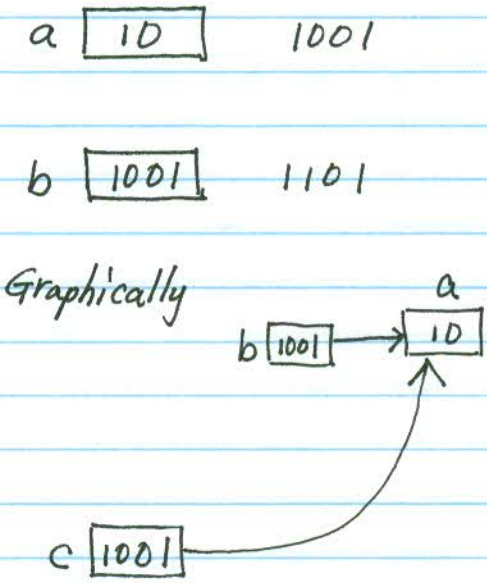
by value: a | 10

```
void f(int b)               b = a
{                           b | ~~10~~ 20
   b = 20
}
```

pointer Variables

A pointer variable stores the memory address of another variable.

```
int a =10;
int * b = &a;
int * c;
c = &a;
```



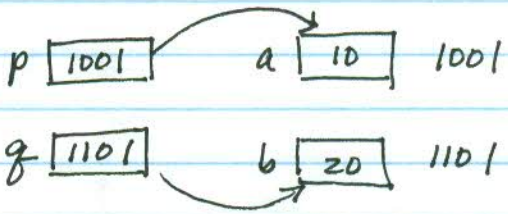
Indirect access / change

```
*b = 30
```

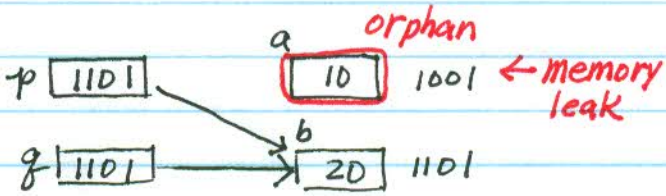
* de-reference operator

a ~~10~~ 30

```
int a =10
int b =20
int * p
int * q
p = &a
q = &b
```

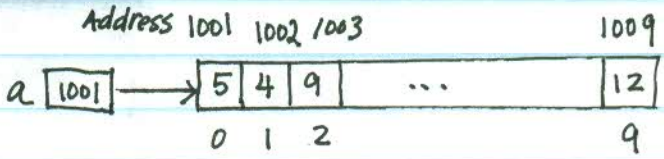


```
p = q
```



Relationship between pointers and Arrays

```
int a[10];
```



Variable a is a pointer variable pointing to the first element of array a.

Third element `a[2]` ⟺ `*(a+2)`

(9)

`a+2`: address of the third element

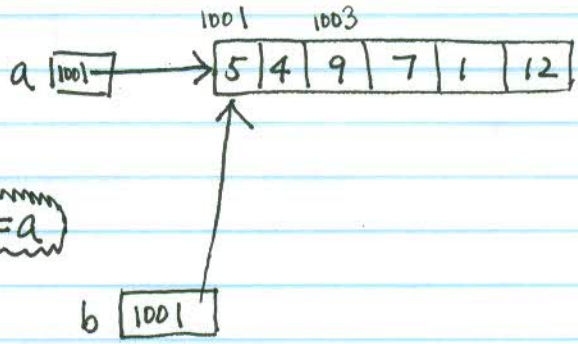
`*`: dereference operator

```
cout << a[2];
cout << *(a+2);
```

e.g.

```
int main ( )
{
    int a[] = {5, 4, 9, 7, 1, 12}
    f(a);
}

void f(int b[])
{
    b[2] = 10;
}
```



`b[2]` → `*(b+2)`

Variable a passes the memory address of the first element of array a to variable b. Two pointer variables a & b point to the same array.