Pointers

What is a pointer ??

Basically, Pointers are variables that stores the address of other variables.

Let's see an example :-

```
int main(){
    int a = 10;
    int *aptr;
    aptr = &a;      // To store the address of a in aptr
   cout << &a << endl;   // 2000
   cout << aptr << endl;   // 2000
   cout << *aptr << endl;   // 10
   return 0;
}
```

$*aptr \rightarrow$ de-referencing of pointers.

Memory

2000	a = 10
4000	aptr = 2000

Pointer Arithmetic

Let's see by examples

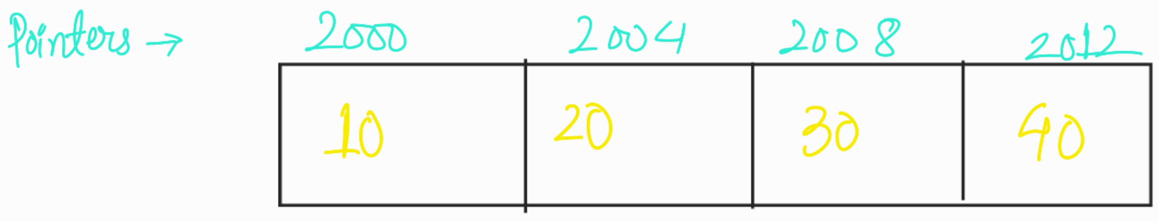
```
int main() {
    int a = 10;
    int *aptr = &a;        stores address of 'a'.
    Cout << aptr << endl; // 2080
    aptr ++;  // 2004      as int takes 4 byte so, it will get increment by 4 each time
    cout << aptr << endl;
    return 0;
}
```

Should be same

```
int main() {
    char ch = 'a';
    char << cptr << endl;
    cptr ++; // 2001       becz char takes 1 byte memory. So each time, it will be incremented by 1.
    cout << cptr << endl;
    return 0;
}
```

Pointers and Arrays

int arr[] = [10, 20, 30, 40]



Let see a Q : To print each element of array using the concept of pointers.

```
int arr[] = {10, 20, 30};
Cout << *arr << endl;  // 10
int  *ptr = arr;
for (int i = 0; i < 3; i++) {
    cout << * ptr << endl;      OR ⟶ Cout << *(arr+i) << endl;
    ptr ++;
}
```

Pointers to Pointers

E.g

```
int main() {
  int a=10;                 2000
  int *p;  → initializing pointer(*)
  p=&a;  → stores the address of a      4000
  cout<<*p<<endl; //10      4200
        de-referencing 'a'
  int **q = &p; → stores the address of p.
  cout << *q << endl; //2000
        de-referencing 'p'
  cout << **q << endl; //10
  return 0;
}
```

This is known as referencing.

de-referencing 2 times (as ** is used)

Memory

a=10

p= 2000

q = 4000

Passing pointers to function...

```
void increment (int a){
      a++;
}

int main() {
   int a=2;
   inreament( a) ;
  cout <<a <<endl ; //2
 return 0;
}
```

Different variable 'a'

value will remain unchanged

calling function

Because, we know from function that, we need to pass values as a parameter and here, 'a' are two different variables.

E.g → Swap using pointers

**(Also, known as ~~Calling by reference~~)

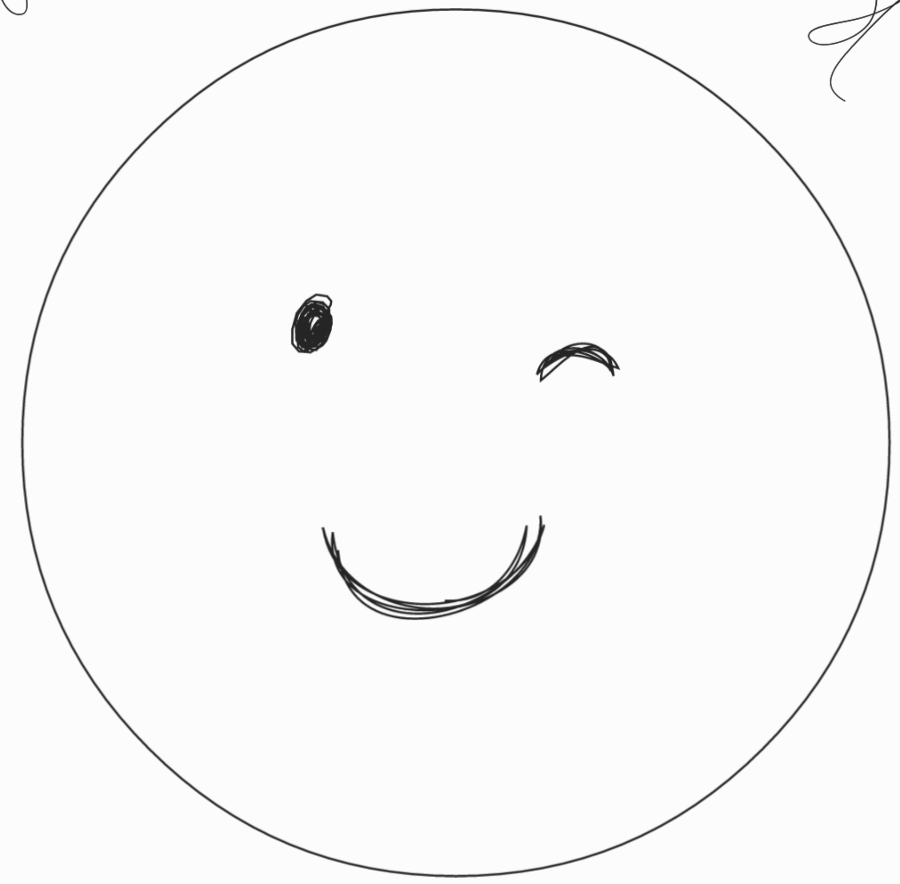
```
Void swap(int *a , int *b){
    int temp = *a;
    *a  =  *b;
   *b = temp;
}
int main () {
   int a=2, b=4;
  swap( &a , &b);
 cout << a<<"  " <<b <<endl;
 return 0 ;
}
```

As on calling the function, addresses of 'a' and 'b' are passed.
So, → Here initialisation of pointer is required.

Sending directly the addresses of a and b in the calling function.

otherwise, it won't work if we give only a and b.

Always Keep y [illegible]



Keep learning!!
DREAMS ON