Review
Pointers in C

Lab 1

What are Pointers?

- A pointer is a variable that store the address of another variable, i.e., memory location.

Pointer Operator in C Program:

Operator	Operator Name	Purpose
* (asterisk/star)	Value at Operator	Gives Value stored at Particular **address**
& (ampersand)	Address Operator	Gives Address of **Variable**

Source: http://www.c4learn.com/c-programming/c-pointer-operator/

Pointer Declaration

What Does following declaration tells compiler ?

```
int *ptr;
```

1. 'ptr' is declared as that 'ptr' will be used only for storing the address of the integer valued variables
2. We can also say that 'ptr' points to integer
3. Value at the address contained in 'ptr' is integer .

Data type:
- int *
- char *
- float *
- …

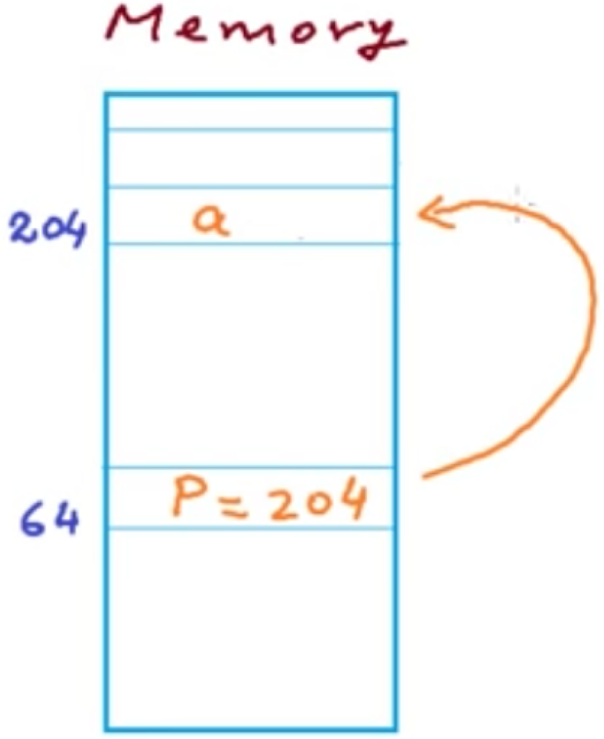
Source: http://www.c4learn.com/c-programming/c-pointer-operator/

To store the address ‘a’ in pointer ‘p’

Introduction to pointers in C

Pointers - variables that store address of another variable

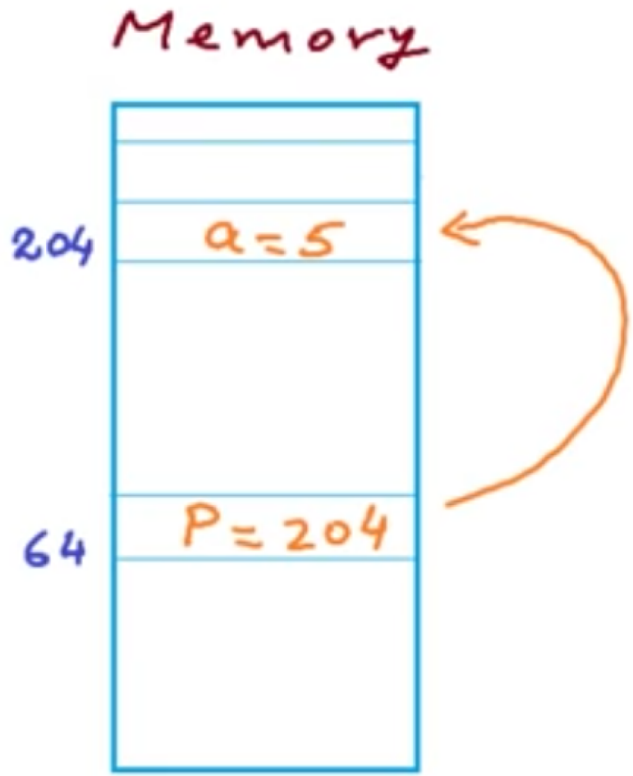
```
int a;
int *p;
p = &a;
```



Source: https://youtu.be/h-HBipu_1P0

Operator	Purpose
*	Gives Value stored at Particular address
&	Gives Address of Variable

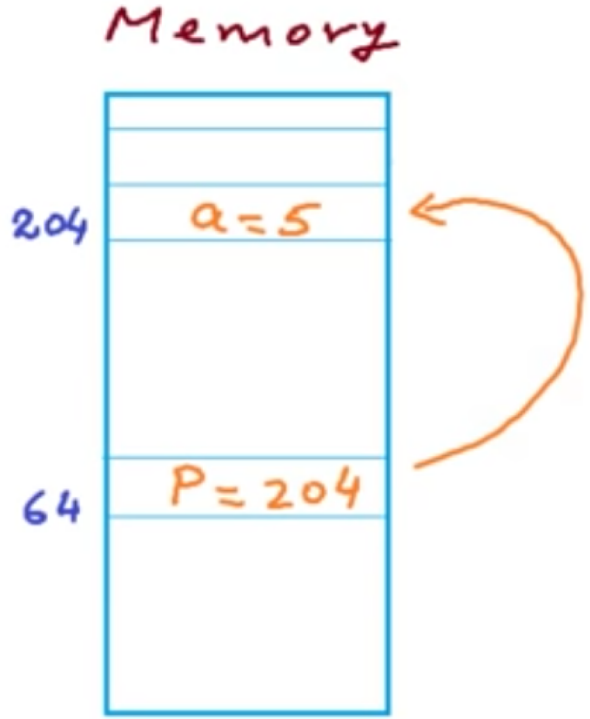
```
int a;
int *p;
p = &a;
a = 5;
Print p   // 204
Print &a  // 204
Print &p  // 64
```



Operator	Purpose
*	Gives Value stored at Particular address
&	Gives Address of Variable

Pointers - variables that store address of another variable

```
int a;
int *p;
p = &a;
a = 5;
Print p     // 204
Print &a    // 204
Print &p    // 64
print *p    // 5   ⇒ derefrencing
```



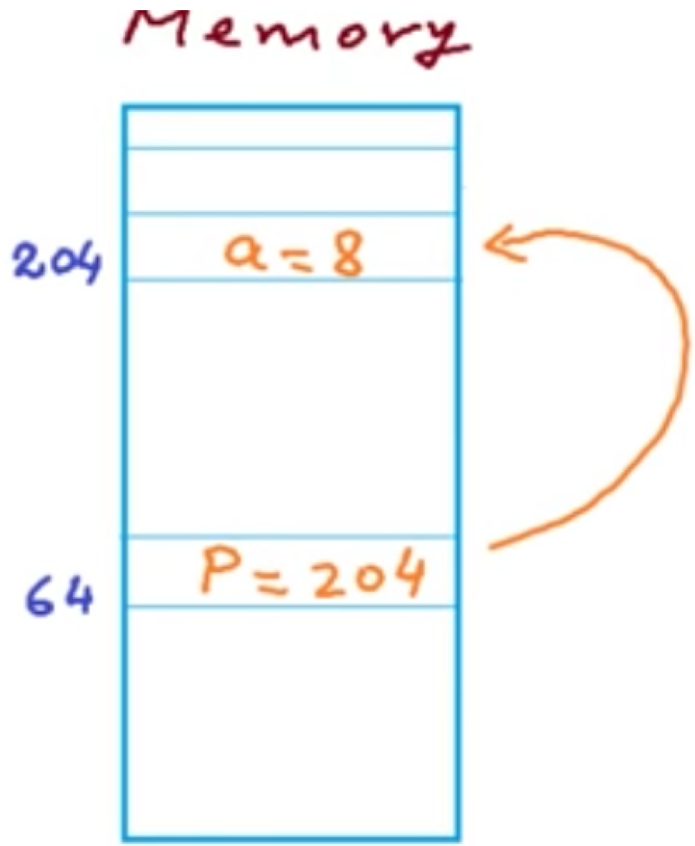
What is Dereferencing Pointer ?

1. Dereferencing Operation is performed to access or manipulate data contained in memory location pointed to by a pointer
2. Any Operation performed on the **de-referenced pointer** directly affects the value of variable it pointes to.

Source: http://www.c4learn.com/c-programming/c-dereferencing-pointer/

Operator	Purpose
*	manipulate Value stored at Particular address

```
int a;
int *p;
p = &a;
a = 5;
Print p      // 204
Print &a     // 204
Print &p     // 64
print *p     // 5
*p = 8
Print a      // 8
```



Print *p;