

# Referencing and Deferencing

By: Annamalai A

---

## Memory Address

A memory address is like a unique identifier for each variable's location in the RAM. Each memory cell stores a small, fixed size of data which is typically one byte. Every variable in `C` is stored in a specific address.

The **address-of** (`&`) operator is used to get the address of a variable. To print the address of a variable use the `%p` format specifier. Using anything else prints garbage value.

## Pointer (`*`)

It is a special variable that stores the address of another variable. Instead of holding the value stored by the variable, it points to the address where the variable is present.

It is very useful in places where we need to access the value of a variable without creating copies.

Here is the syntax to define pointers: `datatype *pointer_name`

```
int a = 10;  
int *p;  
p = &a;
```

`p` → gives the address of `a`.

`*p` → gives the value stored at that address.

A pointer requires datatypes so that the compiler knows how much memory it should occupy and what data type it can actually retrieve from the address.

## Referencing

Referencing is the process of **converting a variable into its address** so that something else (like a pointer or function) can work with that location.

Since every variable occupies a memory address, referencing simply means:

1. Take a variable
2. Get its address using &
3. Store that address somewhere (usually in a pointer)

This process doesn't change the variable's value, rather it only reveals *where* it lives in memory.

Think of referencing as giving someone the **house address** instead of the person inside the house.

Here is an example of referencing:

```
p = &a; // referencing
```

## Dereferencing

Dereferencing is the next step after referencing. It is how you **use** the address stored in a pointer. The process is:

1. Take the pointer that holds an address
2. Follow that address to the variable it points to
3. Read or modify the value stored there

So dereferencing is about **accessing data using an address**.

```
*p = 20; // dereferencing to modify the value at &a
```