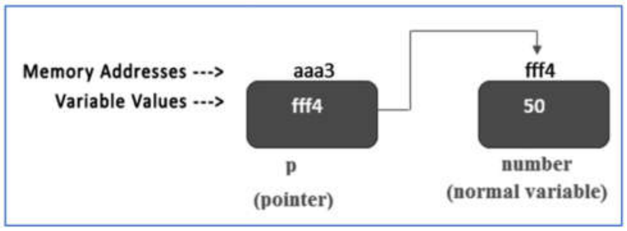
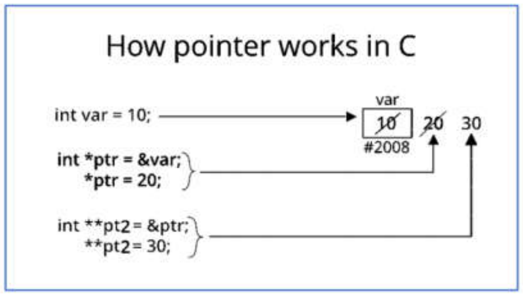
# COMP301 Fall 2022: Lab 3: C Pointer Basics





~~~~~~~~~~~~~~~Sample Code (compile, run and understand the following code)~~~~~~~~~~~~~~~~~~~~~

```
#include<stdio.h>
int func(int *e, char *f, char d[]); //function prototype. function declared at the end


int main(int argc, char*argv[]){
  //1. define a pointer, assign a pointer
  int x=255;
  int *i;                            //defining and declaring i as a pointer
  i = &x;                            //pointer assignment, now i points to x
  char c='r';
  char *p;
  p=&c;
  printf("\nValue of x is%d: ",*i); //print value of int type variable using pointer
  printf("\nValue of c is%c: ",*p); //print value of char type variable using pointer
  printf("\nAddress of x in Memory is: %ld",(long int)&x);
  printf("\nAddress of x in Memory is: %ld",(long int)i);
```

//What are the two purposes of a * when written right before the name of a variable?

→ first purpose ( to declare/initialise a pointer variable)

→ Second purpose ( to dereference a pointer, display value at that pointing address)

//What is the meaning and purpose of a & (the ampersand sign) when written right before the name of a variable?

→ meaning & purpose of & sign is to display the pointer address of a variable. (reference of variable) {& and * are opposite)

//What are the names of the * and & operators when used as in the example above.

* → dereferencing operator & referencing operator/ampersand
~~~~~~~~~~~~~~~

```
  //2. arrays: char array as string, scanf(), name of array is a pointer
  char name[100];
  printf("\nEnter your full name: ");
  scanf("%[^\n]",name);              //this format specifier is used instead of %s
                                     //because it allows to have input with spaces
                                     //between them
  printf("\nYour full name is: %s",name);//%s expects a memory address after comma
```

//The name of the array holds the memory address to the starting element of that array in memory, is that correct? Then what would $*array$ refer to?

*array would dereference and show us the value at the starting index of array.

//What would be the output of $*name[0]$ and why does the output make sense?

```
  //3. function parameter passing by pointer
  func(i, p, name);
```

//Are we passing a list of values to the function or are we sending single variables, no matter what type?

Single variables.

```
  //4. command line arguments(CLA): argv argument address values, argc argument count
  printf("\nThe user entered %d number of CLAs", argc);
  //run the program like    prompt$ ./lab3 a b    to see the effect of above line, a and
  //b are the command line arguments in this example
  int g=1          // and then run like $ ./lab3 a b c to as another example
  while(g<argc+1) {
    printf("\nCLA number %s is",argv[g]);
    printf("\nMemory address of CLA is%ld",(long int)argv[g]);
    g++;
  }
  printf("\n\n"); //two new lines for space
  return 0;
}


int func(int *e, char *f, char d[]){
  printf("\n1st value passed to function is %d",*e);
  printf("\n2nd value passed to function is %s",d);
}
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**Task 1:**

- Write a C program: Use the argv[] to accept the user's age, last name, and then first name, into the program in the same order. The program displays these in the following format:
- For example the user runs: **`$ ./lab3 Khokhar Abdullah 33`**
- The output should be: **`Mr Abdullah Khokhar, is 33 years old`**
- Also print the memory addresses of where first-name is stored in Memory.