- To access address of a variable to a pointer, we use the unary operator **&** (ampersand) that returns the address of that variable. For example &x gives us address of variable x.

```c
// The output of this program can be different
// in different runs. Note that the program
// prints address of a variable and a variable
// can be assigned different address in different
// runs.
#include <stdio.h>

int main()
{
    int x;

    // Prints address of x
    printf("%p", &x);

    return 0;
}
```

- One more operator is **unary** * (Astrik) which is used for two things :
  - To declare a pointer variable: When a pointer variable is declared in C/C++, there must a * before its name.

```c
// C program to demonstrate declaration of
// pointer variables.
#include <stdio.h>
int main()
{
    int x = 10;

    // 1) Since there is * in declaration, ptr
    // becomes a pointer varaible (a variable
    // that stores address of another variable)
    // 2) Since there is int before *, ptr is
    // pointer to an integer type variable
    int *ptr;

    // & operator before x is used to get address
    // of x. The address of x is assigned to ptr.
    ptr = &x;

    return 0;
}
```

  - To access the value stored in the address we use the unary operator (*) that returns the value of the variable located at the address specified by its operand.

```c
// C program to demonstrate use of * for pointers in C
#include <stdio.h>

int main()
{
    // A normal integer variable
    int Var = 10;

    // A pointer variable that holds address of var.
    int *ptr = &Var;

    // This line prints value at address stored in ptr.
    // Value stored is value of variable "var"
    printf("Value of Var = %d\n", *ptr);
```

```
    // The output of this line may be different in different
    // runs even on same machine.
    printf("Address of Var = %p\n", ptr);

    // We can also use ptr as lvalue (Left hand
    // side of assignment)
    *ptr = 20; // Value at address is now 20

    // This prints 20
    printf("After doing *ptr = 20, *ptr is %d\n", *ptr);

    return 0;
}
```

Output :

```
Value of Var = 10
Address of Var = 0x7fffa057dd4
After doing *ptr = 20, *ptr is 20
```

Below is pictorial representation of above program: