

Practical 1

Introduction to Pointers and structures using C

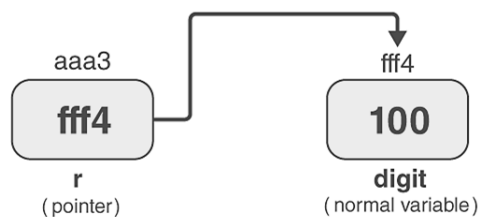
We can use variables to hold address of a memory location and such variables are known as pointer variables. Normal variables are used to store data items of a particular data type (char, int, float etc). Before using a variable in a program, we declare it at the beginning. Similarly we need to declare a pointer variable too in a special way – to let the compiler know we have declared a variable as a pointer (not as a normal variable). To do this we have the * operator – known as *indirection* operator in C.

Pointer variable declaration

The syntax to declare a pointer variable is

(data type) *(variable name);

Ex:- int *ptr ;



It is possible to assign address of single variable or that of an array or the address of a structure etc to a pointer variable.

Reference And De-Reference Operators

Before diving deeper into the concept of pointer let's understand some basics that will help us later on. While using pointers you will definitely use '&' and '*' operators. Now is the time to understand their meaning and use.

First, let's understand Reference operator often called as 'Address of' operator. Using (ampersand) operator with a variable returns us a memory location also known as the address of the given variable.

Example

```
int *ptr;
```

```
int a;
```

```
ptr = &a;
```

The address of variable 'a' is stored in variable ptr.

Now, let's understand the De-Referencing or 'Value at' operator which is denoted by an asterisk (*). It helps in retrieving the value from the memory location which is stored in the pointer variable

Example

```
int *ptr;  
int a;  
*ptr = &a;  
printf("Value of a = %dn", *ptr);
```

Exercise

- 1) Define a structure which has members that include name, idno and marks. Write a C program to read the information about N students and print the name and idno of the students having marks less than M.
- 2) Write a C program that takes Student structure array [for example struct Student st[10] defines an array called student that consists of 10 elements. Each element is defined to be of the type struct Student]. The program should store marks of 5 subjects and calculate total and percentage. Display each subject marks, total and percentage for all students.
- 3) Write a C program to create a number variable. Create a pointer variable for this number variable. Create another pointer-to-pointer variable. Display the address and value of all the variables including pointer variables.
- 4) Write a C program to create an array of 15 elements. create pointer which points to an array. Now print the base address of the array. Then display the array elements using pointer arithmetic.
- 5) Write a program to create a value array and another pointer array. Both array size should be 5. Now store some values in the value array. Then store the address of these five elements in the pointer array. Then print the address and value of each of the pointer array.
- 6) Write a C program to swap 2 numbers using pass by value and pass by reference to a function.