Pointers

Pointers are variables that store the address of other variables.

Syntax

```
//Syntax
//<data_type> *name_of_ptr
int *myIntPointer;
char *myCharPointer;
```

Getting started with the Pointers

Every variable is stored in the memory and each memory location has its own memory address. It enables us to pass variables by reference.

'&' Operator: It gives the address of the variable.
'*' Operator: It gives the value stored at the address, i.e dereferences the value stored at the address

Working Demo

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

cout << *aptr << endl;
*aptr = 20;
cout << a << endl;
```

In the above example,

1. We declared an integer variable 'a', and an integer pointer 'aptr'.
2. The address of 'a' (&a) is stored in aptr.
3. cout << *aptr; gives the value stored at the memory address.
4. *aptr = 20 changes the value at the memory address to 20. Value of 'a' gets changed to 20.

Array Pointers

In C++, The name of the array is a pointer that points to the first element of the array.

```
int arr[] = {10, 20, 30};
cout << *arr << endl;

int *ptr = arr;
for (int i = 0; i < 3; i++) {
    cout << *(arr + i) << endl;
}
```


*(arr + i) is equivalent to arr[i].

(arr + i) is the address of the ith element of the array.

Swapping 2 variables using the address

Address of ‘a’ and ‘b’ gets interchanged, thus their values get swapped.

```
int a = 2;
int b = 4;

swap(&a, &b);
cout << a << " " << b << endl;
```


Supplementary material:

Output-based questions/ MCQs.