C++ Arrays

→ Array is a group of similar types of element that have contiguous memory location.
→ Array index starts from zero.
→ We can store only fixed set of elements.

Data →	10	20	30	40
Address →	0	1	2	3

Advantages
→ Code optimize
→ Random Access
→ Easy to traverse
→ Easy to manipulate of sort data.

Disadvantage : Fixed size.
→ Insertion of deletion are costly.

Note: Name of array access to the or point to the first element of array.

Array Type
→ There are two types of Array
1. Single Dimension Array.
int a[2] = {1,2,3};

2. Multidimension Array
r c
int a[2][2] = {1,2,3,4,5,6};
00 01 02 10 11 12 20 21 22
{1 2 3},
{4 0 3},

Passing Array to function.

To reuse the array logic, we can create function. To pass array to function, we need to provide only name.

ex: functionname(array name); // Syntax

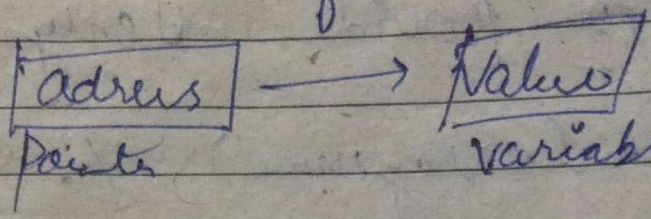
ex:
```
#include<iostream>
using namespace std;
void printArray(int arr[5]);
int main()
{
    int arr1[5] = {10, 20, 30, 40, 50};
    int arr2[5] = {5, 15, 25, 35, 45};
    printArray(arr1); printArray(arr2);
}

void printArray(int arr[5])
{
    cout<<"Printing array elements:" endl;
    for(int i=0; i<5; i++)
    {
        cout<<arr[i]<<"\n";
    }
}
```

Here we pass the array!

Pointer

→ The pointer is indicater that points to an address of a value.



Advantage:
→ reduce the code
→ improve performance
→ used with arrays, structure of function

Usage of pointer
i → Dynamic memory allocation.
→ We can dynamically allocate memory using malloc() of calloc() function where pointer is used.

ii → Array, function of structure

Symbol used in pointer
& → Address operator : for address of Var
* → Indirect operator : for Value of addr

Declaration
int *a;
char *a;

eg *p = &a

Array of pointer

→ An array of pointer is an array that consists of variables of pointer type, which means that the variable is a pointer addressing to some other element.

ex: int *ptr[5];

→ This is called a pointer Array or Array of pointer name ptr.

→ The element of an array of a pointer can also be initialize by assigning the address of some other elements.

ex: int a;
ptr[2] = &a;

here in array at index 2 there is address of a variable

Array of pointer to strings

• Difference Array of pointer to the Array of pointer to string

→ An array of pointer to string is more efficient than the two dimensional array of characters. In case of memory consumption.

→ In an array of string pointer, the manipulation of string is comparatively easier than in the case of 2 d array.

→ We can also easily change the pointers using the pointers.

Void Pointer

→ A void pointer is a general-purpose pointer that can hold the address of any data-type, but it is not associated with any data type.

Note: We cannot assign the address of a variable to the variable of a different data type i.e int can't address float etc so on.

Note: If we want to hold the address of any data type we use the void-pointer,

Void

Difference b/w C & C++ void pointer

→ In C we can assign the void pointer to any other pointer type without any typecasting

whereas in C++ we need to typecast when we assign the void pointer type to any other type.

* int a=10
* &a ⇒ value deya!

Reference (Address)

Reference can be created by using &(ampersand operator)
It occupies some memory locations.
Therefore we can access the original variable by using either name of the variable or reference of the variable.

ex: int a=10;
int &ref = a; } same thing.
value deya!

Note: Reference variable cannot be modified.

Reference can also be passed as a function parameter. It does not create a copy of the argument it behaves as an alias for a parameter.

Key point :→ It enhance the performance as it does not create a copy of the argument.

Key point :→ With the help of reference we can access the nested loop data.

Key point → We cannot assign the NULL value to the reference variable, but the pointer variable can be assign with a NULL value.

Function Pointer

→ The function pointer is a pointer use to point functions.
→ It is basically used to store the address of function
→ We can call the funct by using the pointer function
→ They are mainly useful for event-driven application, callbacks or used for storing the function in array.

* Syntax: int (*FunPtr)(int val);

Address of a function:

→ For getting func address, we just need to mention the name of the funct. we do not need to call the funct

Passing a function pointer as a parameter

→ The function pointer can be passed as a parameter to another function.

Eg:
```
#include <iostream>
using namespace std;
void func1()
{
}
void func2 (void (*funcptr)())
{
  funcptr();
}
int main()
{
  func2(func1)
  return 0;
}
```

Memory Management :

Memory management is a process of managing memory, assigning the memory space to the programs improve the overall system process

To avoid the wastage of memory, we use the new operator to allocate the memory dynamically at run time.

Memory Management Operators.

We use the malloc() or calloc() function to allocate the memory dynamically at run time, Free() is used to deallocate the dynamically allocated memory.

C++ also defines unary operator such as new of delete.

New operator :

→ A new operator is used to create the object ; while a delete operator is used to delete the object

New operator exists lifetime till we delete it. It is independent of block.

Syntax :- pointer_variable = new data-type

* int = new int;

ex: int *p;
q = new int.

delete pointer_variable;
delete p;