**Aman**

1 What are pointers

In computer science, a pointer is an object in many programming languages that stores a memory address. This can be that of another value located in computer memory, or in some cases, that of memory-mapped computer hardware. A pointer references a location in memory, and obtaining the value stored at that location is known as dereferencing the pointer. As an analogy, a page number in a book's index could be considered a pointer to the corresponding page; dereferencing such a pointer would be done by flipping to the page with the given page number and reading the text found on that page. The actual format and content of a pointer variable is dependent on the underlying computer architecture.

**Shardul**

2 Pointers in C/C++

As we already know, after declaring a variable in the C/C++ programming language, the compiler automatically allocates a reserved amount of memory for the variable depending on its data type.

For example, a 64-bit compiler reserves 8 bytes of memory for the double data type. Now, this variable has a memory location/address associated with it represented by a numeral through which can be accessed. Just like you have a house address that tells your location, a variable also has an address that tells you where the computer is storing it in its memory.

Whenever we use the scanf() statement, we use the ampersand symbol (&), called the reference. This reference operator indicates that the memory address is being allocated to the variable.

Pointer in C and C++ is nothing but a way to access a variable by storing its memory location. In programming terminology, A pointer is simply a variable that stores the memory location of another variable.

**Sanat**

3 Significance of Pointer in C/C++

We can easily access the memory location of any variable directly and manipulate it according to our convenience.

It facilitates the concept of dynamic memory allocation, making it a much faster language as compared to programming languages that do not support the use of pointers like Python and Java.

We prefer pointers in certain situations to improve the efficiency of certain procedures

It reduces length of the program and its execution time as well. It allows C language to support Dynamic Memory management.

**Shreerang**

4 Dynamic Memory Allocation using Pointers

C offers the implementation of dynamic memory allocation with the help of pointers

In the C programming language, memory can either be allocated statically or dynamically

Just like static memory allocation is done with the help of arrays when the amount of memory to be allocated is known beforehand, dynamic memory allocation is done with pointers when the amount of memory to be allocated is unknown.

There are basically two keywords helpful while working with DMA:

new: We use it to allocate memory dynamically and to return a pointer that stores the memory address of the variable or data structure.

delete: We use it for the deallocation of memory that is pointed by the pointer.

**Sushant**

Intro conclusion:

A pointer is a simple, more concrete implementation of the more abstract reference data type. Several languages, especially low-level languages, support some type of pointer, although some have more restrictions on their use than others.

While "pointer" has been used to refer to references in general, it more properly applies to data structures whose interface explicitly allows the pointer to be manipulated (arithmetically via pointer arithmetic) as a memory address, as opposed to a magic cookie or capability which does not allow such.[citation needed] Because pointers allow both protected and unprotected access to memory addresses, there are risks associated with using them, particularly in the latter case. Primitive pointers are often stored in a format similar to an intege