Void Pointer in C

Void pointer in C is a unique type of pointer that can point to any object's data type. A void pointer in C is a general-purpose pointer that can hold any data type's address and can be typecasted to any type. This type of pointer is essential for the GATE exam. A Void pointer has no datatype associated with it.

With such void pointers, memory allocation becomes very simple. Because they make the functions flexible enough to become appropriately allocated with bytes and memories, let's look at the C syntax for the void pointer.

Syntax of Void Pointer in C

void *pointer name;

The void keyword serves as the pointer type. In this case, it is followed by the pointer name to which the pointer type points and allocates the address location in the programming code. The name and type of pointer that supports any given data type are declared when a pointer is declared.

void *p

The pointer expects a void type, not an int, float, or anything else. It is indicated by the p pointer, which includes the * symbol. It indicates that this pointer has been declared and will be denoted in the future.

Size of Void Pointer in C

In the C programming language, the size of the void pointer is the same as the size of the character datatype pointer. The C language standard states that a pointer to void has the same representation as a pointer to a character datatype. The pointer size will differ depending on the platform we're using. Please see the example below for a better understanding.

```
#include <stdio.h>

int main ()

{

    void *VP = NULL;  //Void Pointer

    int *IP = NULL;  //Integer Pointer

    char *CP = NULL;  //Character Pointer

    float *FP = NULL;  //Float Pointer
```

```
    printf ("Size of Void Pointer = %d\n\n", sizeof (VP));

    printf ("Size of Integer Pointer  = %d\n\n", sizeof (IP));

    printf ("Size of Character Pointer = %d\n\n", sizeof (CP));

    printf ("Size of Float Pointer = %d\n\n", sizeof (FP));

    return 0;

}
```

Output

Size of Void pointer = 8

Size of Integer pointer = 8

Size of Character pointer = 8

Size of Float pointer = 8

Why is Void Pointer in C useful?

One major advantage of the void pointers in C is that they can be 'reused'. Due to this characteristic of the void pointer in C, the complexities of 'space' and 'time' can be reduced to a certain extent, which is very important in Embedded System design. Void Pointer in C is also used to formulate the questions in the GATE CSE question paper. They are very helpful when you want a single pointer to reference data of various types at various times.

Please take a look at the following simple code snippet to show the following simple code snippet to demonstrate how void pointers can be reused and thus reduce space complexity.

```
#include <stdio.h>

int main()

{

int I = 10;

char c = 'A';

float f = 12.34;

double d = 43.78;
```

```
//Declaring a generic pointer

void *vptr;

//Storing the address of the integer variable

vptr = &i;

printf("The value of t: %d\n", *(int*)vptr);

//storing the address of the character variable

vptr = &c;

printf("The value of c: %c\n", *(char*)vptr);

//storing the address of the float variable

vptr = &f;

printf("The value of f: %f\n", *(float*)vptr);

//storing the address of the double variable

vptr = &d;

printf("The value of d: %f\n", *(double*)vptr);

return 0;

}
```

Output

```
The value of t: 10

The value of c: A

The value of f: 12.340000

The value of d: 43.780000
```

Advantages of Void Pointer in C

The functions malloc() and calloc() return the void* type (void pointers). This enables us to use these methods to allocate memory of any data type. Candidates can expect the questions in the GATE CSE exam on various advantages of the void pointer in C.

Void pointers in C are used to implement generic functions in the C programming language. For instance, the qsort() function in the standard C library.