* void Pointers in C++.

A void pointer in C++ is a pointer that can point to any data type. void Pointer are also known as Generic Pointer because they can be used to point to any type of object or data. void Pointers are declared using the keyword "void" as the pointer type.

syntax:-

```
void *ptr;
```

This pointer can be used to point to any type of data, but we cannot deference it directly since the compiler does not know the datatype it is pointing to.

To use a void pointer, we must first cast it to a specific datatype. For example - if we want to use a void pointer to point to an integer, we can cast it to an int* datatype.

```
void *ptr;
int x = 10;
ptr = &x;
int *intPtr = static_cast<int*>(ptr);
cout << *intPtr;
```

"static_cast" operator is used to cast the void pointer to an int* pointer. It tells the compiler that the void

Date.................

pointer should be interpreted as a pointer to an integer.

void pointer are commonly used in c++ for Dynamic Memory allocation using the new operator. when we allocate memory dynamically using new, the type of the allocated memory is determined at runtime. we can use a void pointer to the dynamically allocated memory & then cast it to the appropriate datatype when we need to use it.

* static_cast operator:

In C++, static_cast is a casting opera
-tor that is used to convert a value
from one datatype to another.
Syntax:-

```
static_cast<new_type>(expression);
```

"new_type" is the datatype in which we
want to cast the expression. The
expression can be a variable, a lite
-ral or any valid expression in C++.

Note:- static_cast is a compile time
operator, it means that the conversi
-on is done at compile time rather
than at runtime. This can result in
faster code execution since the compi
-ler can optimize the code based on
the specific datatype that are being
used. However, it also means that the
cast can fail at runtime if the data
types are not compatible.