NULL Pointer

In C++, a null pointer is a special pointer value that indicates that the pointer is not pointing to any valid memory location. It is typically represented by the constant value **`nullptr`** or **`NULL`** and is used to signify that the pointer is not intended to point to any valid entity.

Here are some key points about null pointers in C++:

1. **nullptr**: In C++, the keyword **`nullptr`** or **`NULL`** is used to represent a null pointer.
2. **Pointer Initialization**: You can initialize a pointer to be a null pointer by assigning it the value `nullptr`. For example:

   ```
   int *ptr = nullptr; // ptr is now a null pointer
   ```

3. **Checking for Null Pointers**: Before dereferencing a pointer, it's important to check if it's a null pointer to avoid runtime errors (such as segmentation faults). This can be done using an `if` statement or by directly checking the pointer value.

   ```
   int *ptr = nullptr;
   if (ptr != nullptr) {
       // Pointer is not null, safe to dereference
       *ptr = 10;
   }
   ```

4. **Pointer Arithmetic**: Null pointers can participate in pointer arithmetic, but attempting to dereference a null pointer results in undefined behavior. It's crucial to ensure that a pointer is not null before attempting to dereference it.

   ```
   int *ptr = nullptr;
   ptr++; // Valid, but ptr is still a null pointer
   ```

5. **Default Initialization**: If a pointer is declared without initialization, it will have an indeterminate value, which is not guaranteed to be a null pointer. It's good