Variabel pointer tidak berbeda jauh dengan variabel biasa. Ketika kita membuat variabel pointer dan tidak memberi nilai awal di saat itu maka kita akan mendapatkan nilai sampah pada variabel pointer yang kita dirikan.

Penulis sering mengatakan bahwa inisialisasi itu penting tapi bukan berarti kalian harus inisialisasi menggunakan nilai yang bernilai. Ada suatu kata kunci yang dapat anda gunakan untuk menginisialisasi dengan nilai yang kosong, kata kunci itu adalah NULL.

NULL adalah Preprocessor macro berasal dari Bahasa C yang di dalamnya berisi nilai 0. NULL sangat berguna untuk mengkosongkan nilai dari suatu variabel.

#include <iostream>

int main()

{

int \*p=NULL;

std::cout<<p;

return 0;

}

NULL tidak hanya untuk digunakan pada variabel pointer,kata kunci NULL juga dimungkinkan digunakan pada apa saja dan dimanapun itu, dan akan tetap memiliki nilai/fungsi yang sama.

nullptr

tapi untuk memberi nilai kosong pada variabel pointer, NULL bukanlah pilihan tepat untuk melakukan itu. Pada C++ 11 dikenalkan kata kunci baru yang dikhususkan untuk memberi nilai kosong pada pointer, yaitu nullptr.

Nullptr adalah kata kunci konstanta rvalue, mirip dengan nilai Boolean false. C++11 menggantikan NULL dengan nullptr karena NULL sebenarnya lebih ditujukan untuk variabel biasa (bukan pointer), dan NULL memiliki nilai 0 yang merupakan integer. hal itu tidak konsisten dengan wadah yang berupa pointer.

Dan di bawah ini akan membuktikan bahwa null adalah integer dan nullptr pantas untuk nilai pointer.

#include <iostream>

#include <typeinfo>

int main()

{

std::cout<<typeid(NULL).name()<<"\n"<<typeid(nullptr).name();

return 0;

}

Untuk pertama kali berdasarkan pengertian NULL dan nullptr akan terdengar memiliki fungsi yang sama, tapi dalam program di atas akan menunjukan perbedaan dari NULL dan nullptr.

nullptr\_t

nullptr\_t adalah tipe data dari C++11 yang dimungkinkan untuk digunakan sebagai tipe data, dan nullptr\_t hanya dapat menerima satu nilai yaitu nullptr.