**More on References and Pointers**

https://www.youtube.com/watch?v=DvUYbUn9uQ4&list=PL82YdDfxhWsCyZLsg\_kXhH8sy5ixQNras&index=5

#include "stdafx.h"

#include <iostream>

int main()

{

int a[]{1,2,3};

int \*p1 = &a[1];

std::cout << \*p1 << std::endl; //2

std::cout << p1[0] << std::endl; //2

//p1[0] is the same as \*(p1+0)

std::cout << p1[1] << std::endl; //3

std::cout << p1[2] << std::endl; //-874665443

//There's no value at a[3]

std::cout << std::endl;

std::cout << a << std::endl; //006FFA6c

std::cout << p1 << std::endl; //006FFA60

std::cout << p1+1 << std::endl; //006FFA64

std::cout << \*(p1 + 1) << std::endl; //2

// \*(p1+1) is the same as p1[1]

// \*(p1+2) is the same as p1[2]

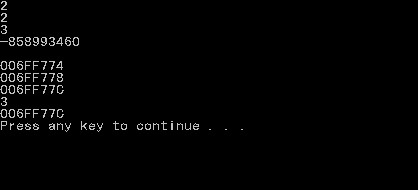
std::cout << &p1[1] << std::endl; //006FFA64

//Everytime it is run, the memory destination changes

//For example, 006FFA64 changed into 00B6F9A0

}

**Result:**



**Important notes:**

* The ‘\*’ symbol is a de-reference operator
* std::cout << p1[0] << std::endl; //2
* //p1[0] is the same as \*(p1+0)
* Right, you can use pointers for arrays as well
* Variables’ memory locations change every time program runs
* Try to understand the code to learn more