**Difference between local and static arrays**

(Code copied from the C++ book)

#include "stdafx.h"

#include <iostream>

#include <array>

void staticArrayInit(); //function prototype

void automaticArrayInit(); //function prototype

const size\_t arraySize{3};

int main()

{

std::cout << "\n////////// First call to each function:";

staticArrayInit();

automaticArrayInit();

std::cout << "\n\n////////// Second call to each function:";

staticArrayInit();

automaticArrayInit();

std::cout << std::endl;

}

//function to demonstrate a static local array

void staticArrayInit(void) {

//initializes elements to 0 first time function is called

static std::array<int, arraySize> array1; //static local array

std::cout << "\n\nValues on entering staticArrayInit: \n";

//Output contents of array1

for (size\_t i{ 0 }; i < array1.size(); ++i) {

std::cout << "array1[" << i << "] = " << array1[i] << " ";

}

std::cout << "\n\nValues on exiting staticArrayInit:\n";

//Modify and output contents of array1

for (size\_t j{ 0 }; j < array1.size(); ++j) {

std::cout << "array1[" << j << "] = " << (array1[j]+=5) << " ";

}

}

void automaticArrayInit(void) {

//Initializes each time this function is called

std::array<int, arraySize> array2{1,2,3}; //automatic local array

std::cout << "\n\nValues on entering automaticArrayInit: \n";

//Output contents of array2

for (size\_t i{ 0 }; i < array2.size(); ++i) {

std::cout << "array2[" << i << "] = " << array2[i] << " ";

}

std::cout << "\n\nValues on exitting automaticArrayInit: \n";

//Modify and output contents of array2

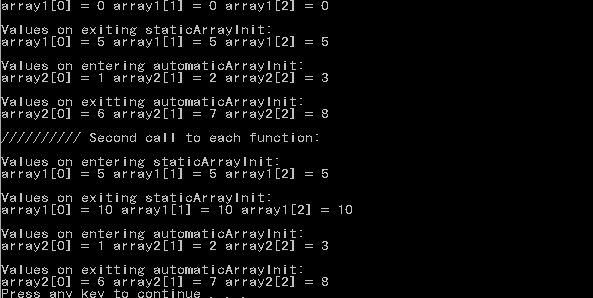
for (size\_t j{ 0 }; j < array2.size(); ++j) {

std::cout << "array2[" << j << "] = " << (array2[j] += 5) << " ";

}

}

**Result:**



**Important notes:**

* Declaring static variables inside functions is more efficient than declaring local (automatic) arrays. Because every time a block starts, an array has to be created and initialized; the local array is then destroyed each time function terminates
* Static arrays’ elements, unlike local arrays’, are automatically initialized to 0 by the compiler when the array is created
* (Unrelated to this topic) A constant variable has to be initialized at the same time it was declared