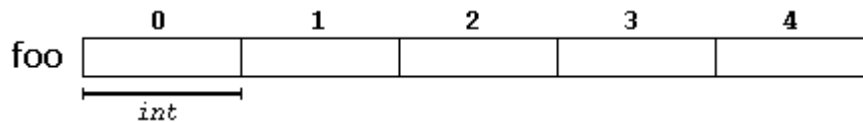


## Section 2: Arrays

Array Definition (Website Reference: <http://www.cplusplus.com/doc/tutorial/arrays/>) : An array is a series of elements of the same type placed in contiguous memory locations that can be individually referenced by adding an index to a unique identifier.

That means that, for example, five values of type `int` can be declared as an array without having to declare 5 different variables (each with its own identifier). Instead, using an array, the five `int` values are stored in contiguous memory locations, and all five can be accessed using the same identifier, with the proper index.

For example, an array containing 5 integer values of type `int` called `foo` could be represented as:



Note: an Array always starts a placement zero. This should be taken into account when filling the array using a loop or user input. Note: an Array always starts an index placement zero. This should be taken into account when filling the array using a loop or user input. So if I have a data set of 20 the first element is located at index 0 and the last element is located at index 19.

### Two basic types of Arrays 1-dimensional and 2-Dimensional

#### Declaring Arrays

Declaring arrays is very similar to declaring a variable that will be used the program. It is suggested to declare the array at the beginning of the program; similar when declaring global variables.

**Step 1:** Declare what type of input will be accepted into an array (I.E. `int`, `char`, `float`, etc)

**Step 2:** Create a name for the array (create a name that makes sense to what the array is doing; similar to a variable)

**Step 3:** Inside `[ ]` declare the size of the array. Note you can always declare an array larger than needed. The user does not need to fill in all of the array in order for it to work

Example

1-Dimensional Array: `int sample_array [4];`

2- Dimensional Array: `int sample_array [8][8];`

## Filling an Array

There are three options to fill in an array the user will place a value (inputted or predefined at the first available index spot starting at 0 and moving up in sequential order.

**Option 1:** When declaring the array the user may declare what the elements of the array are

```
I.E int my_array [ ] = {1,2,3} //the numbers in the bracket [ ] sets the size of the array. In this case the array size is determined by what it is equal too, which are the values placed in between the braces{ }. In this case the array size is 3 with a data set of 1.2.3
```

**Option 2:** When a user is inputting information the programmer can pick where the input goes in the array

```
I.E: cin >> my_array [3]; // in this scenario the user will input a value and the input will be placed in the 3 spot of the array.
```

**Option 3:** Using a Loop. This typically the most common way to fill an array.

```
I.E: for (int x = 0; x < 6; x++)  
{  
    cout << "Enter a Number:";  
    cin >> my_array [x];  
}
```

## Section 2 Program 1: 1D Arrays

**Directions:** Write the following source code for a 1-Dimensional Array. Answer the questions.