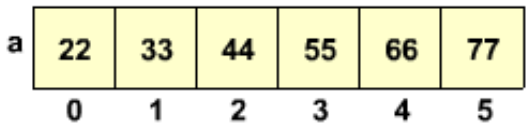
Derived Data Type

Derived data type can be divided into three different categories: (1) Arrays, (2) Function and (3) Pointer. For more information, lets discuss each topic one by one.

Arrays

An array is a collection of one or more identical data items. The objects are called the elements of the array and are numbered consecutively 0, 1, 2, 3, and so on. These numbers are called index values or subscripts of the array. The term "subscript" is used because as a mathematical sequence, an array would be written with subscript: a1,a2,a3, and so on. These numbers locate the element's position within the array, thereby giving direct access into the array.

If the name of the array is a, then a[0] is the name of the element that is in position 0, a[1] is the name of the element that is in position 1, etc.



Lets see how the array named a with 6 elements: a[0] contains 22, a[1] contains 33, a[2] contains 44, a[3] contains 55, a[4] contains 66 and a[5] contains 77.The image actually represents a region of the computer's memory because an array is always stored this way with its elements in a contiguous sequence.

Functions

Functions are the building blocks of C++ programs. A function is a collection of declarations and statements. Each C++ program has at least one function: the main function. The execution of a C++ program begins with main. Additional will be subordinate to main, and perhaps to one another.

If a program contains multiple functions, their definitions may appear in any order, though they must be independent of one another. That is, one function definition cannot be embedded within another.

One of the advantages of using functions is that it is possible to reduce the size of the program by calling and using them at different places in the program. C++ has added many features to functions to make them more reliable and flexible. C++ functions can be overloaded to make it perform different tasks depending upon the arguments passed to it.

Pointers

A pointer is a variable that represents the location rather than the value of a data item, such as a variable or an array element. Pointers are used frequently in C++, as they have a number of useful applications. In C++, pointers are extensively used for memory management and achieving polymorphism. Pointers are also closely associated with arrays and therefore provide an alternate way to access individual array elements.