

C++ provides a data structure, the array, which stores a fixed size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

Instead of declaring individual variables such as `number0`, `number1`, ... `number99`, you declare one array variable such as `numbers` and use `numbers[0]`, `numbers[1]`, and ...`numbers[99]` to represent individual variables. A specific element in an array is accessed by an index.

All arrays consist of contiguous memory location. The lowest address corresponds to the first element and the highest address to the last element.

To declare an array in C++, the programmer specifies the type of the elements and the number of elements required by an array as follows

```
Type arrayName [arraySize];
```

This is called a single-dimension array. The `arraySize` must be an integer constant greater than 0 and type can be any valid C++ data type. For example, to declare a 10-element array called `balance` of type `double`, use this statement

```
Double balance[10];
```

You can initialize C++ array elements either one by one or using a single statement as follows

```
Double balance[5] = {1000.0, 2.0, 3.4, 17.0, 50.0};
```

The number values between braces `{ }` can not be larger than the number of elements that we declare for the array between the square brackets `[]`. Following is an example to assign a single element of the array.

If you omit the size of the array, an array just big enough to hold the initialization is created. Therefore, if you write

```
Double balance [ ] = {1000.0, 2.0, 3.4, 17.0, 50.0};
```

```
Balance [4] = 50.0;
```

The above statement assigns element number 5th in the array a value of 50.0. Array with 4th index will be 5th i.e last element because all arrays have 0 as the index of their first element which is also called base index. Following is the pictorial representation of the same array we discussed above.