**Chapter 7**

Table of Contents

[Array of Objects 2](#_Toc49508359)

[Character Arrays and String 4](#_Toc49508360)

## Array of Objects

We can declare an array of objects like this:

#include<iostream>  
using namespace std;  
  
class ClassName  
{  
 int data;  
  
public:  
  
 void setData(int input)  
 {  
 data = input;  
 }  
  
 int getData()  
 {  
 return data;  
 }  
};  
  
int main()  
{  
 ClassName objArray[5];  
 for (int i=0; i<5; i++)  
 {  
 int num;  
 cin>>num;  
 objArray[i].setData(num);  
 }

for (int i=0; i<5; i++)  
 {  
 cout<<objArray[i].getData();  
 }  
}

C++

Notice that we used a constant size for the array. This is very important. It is poor design to use a variable taken from the user as input as the size of any array, be it for objects or not. While some compilers may allow this, others will indicate errors. The size of the array can be declared with a variable or a macro at the top of the program. In this way, if the size needs to be changed, it can be done in one go instead of having to go through the code and changing every instance of the array size.

## Character Arrays and String

We know that it is possible to store characters in an array. While we generally initialize a string as characters in one go like this:

char example[10] = {“Hello”};

C++

the proper way to initialize a character array is this:

char example[10]={‘H’, ‘e’, ‘l’, ‘l’, ‘o’, ‘\0’};

C++

This is, simply put, irritating. There are other problems with using character arrays. For example, we cannot set multiple values for the array using the = operator except during initialization (we can use the strcpy function though) and the size of the array cannot be changed. Problems like these make character arrays difficult to use with classes.

Instead, we use the string class type:

string example = “Hello”;

C++

This class type is unique in that it dynamically allocates memory. We do not need to provide the length of the string, and if we change the string later on, memory is automatically adjusted.

The usefulness of this class can be seen when trying to concatenate two strings objects. We would need a separate function, strcat, if we wanted to join two character arrays. With string objects, we can do this:

string str1 = “Hello “;  
string str2 = “World!”;  
string str3 = str1 + str2;

C++

Notice that we needed two separate string objects to be able to concatenate them. We cannot directly concatenate two strings.