

Arrays and Strings

C++ Notes

Mrs. Alano

Array - list of items that all have the same type

Subscript - number that indicates the position of the particular array element being used

Element - single object in an array

```
double moneyCollected [5];    //Declaration that reserves
                               //sizeof(double)*5 bytes of memory
```

Array declaration syntax

```
datatype arrayName [size]; //where size must be a constant
```



Figure 5-3 The moneyCollected array in memory

```
int someNumbers [7];
```

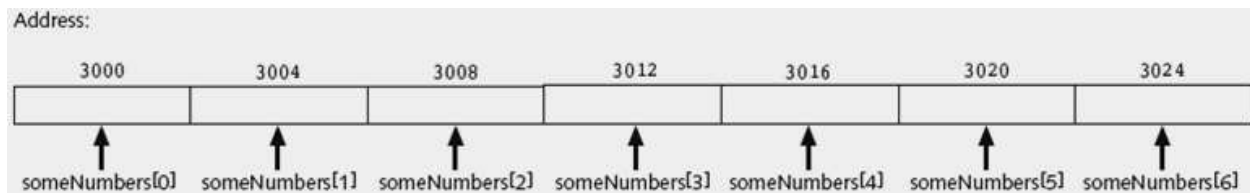


Figure 5-4 The someNumbers array stored at memory address 3000

****NOTE**** If you access `someNumbers[7]`, you may get a warning or a garbage value

Storing Values in an Array

```
int rent[4];
rent[0] = 250;
rent[1] = 375;
rent[2] = 460;
rent[3] = 600;
```

```
int rent[4] = {250, 375, 460, 600};
int rent[] = {250, 375, 460, 600};
int rent[4] = {250, 375};           //rent[2] and rent[3] are set to 0
int rent[3] = {250, 375, 460, 600}; //syntax error
int rent[4] = {0};                 //sets all array elements to 0
```

Accessing and Using Array Values

```
const int SZ_OF_ARRAY = 5;
int arrayInt[SZ_OF_ARRAY] = {34, 56, 12, 3, 99};
for(int x = 0; x < SZ_OF_ARRAY; ++x)
    cout<<arrayInt[x]<<endl;
```

Figure 5-8 Displaying five array values in a loop

```
#include<iostream>
using namespace std;
int main()
{
    const int NUM_PRICES = 10;
    double price[NUM_PRICES];
    int sub;
    for(sub = 0; sub < NUM_PRICES; ++sub)
    {
        cout<<"Enter a price ";
        cin>>price[sub];
    }
    cout<<endl<<"The entered prices, in reverse order:"<<endl;
    for(sub = NUM_PRICES - 1; sub >= 0; --sub)
        cout<<price[sub]<<" ";
    cout<<endl;
}
```

Figure 5-9 Program that allows a user to enter 10 values, then displays them in reverse-entry order

```
#include<iostream>
using namespace std;
int main()
{
    const int NUM_SCORES = 10;
    int score[NUM_SCORES];
    int sub;
    double total = 0;
    double average = 0;
    for(sub = 0; sub < NUM_SCORES; ++sub)
    {
        cout<<"Enter score #"<<(sub + 1)<<" ";
        cin>>score[sub];
    }
    cout<<endl<<"The scores are:"<<endl;
    for(sub = 0; sub < NUM_SCORES; ++sub)
    {
        total += score[sub];
        cout<<score[sub]<<" ";
    }
    cout<<endl;
    average = total / NUM_SCORES;
    cout<<"The average score is "<<average<<endl;
}
```

Figure 5-42 The TestScores.cpp program

Avoiding Common Array Errors

- When working with arrays, common errors include:
 - Forgetting that arrays are zero based
 - Accessing Locations Beyond the Array

Using Part of an Array

```
const int TESTS = 30;
int testScore[TESTS];
int testNum, a;
double total = 0;
double average;
testNum = 0;
cout<<"Enter first test score, or 999 to quit ";
cin>>testScore[testNum];
while(testNum < TESTS && testScore[testNum] != 999)
{
    total += testScore[testNum];
    ++testNum;
    if(testNum < TESTS)
    {
        cout<<"Enter next test score or 999 to quit ";
        cin>>testScore[testNum];
    }
}
cout<<"The entered test scores are: ";
for(a = 0; a < testNum; ++a)
    cout<<testScore[a]<<" ";
average = total / testNum;
cout<<endl<<"The average test score is "<<average<<endl;
```

Figure 5-15 Program that allows user to enter any number of test scores, to 30

Using Parallel Arrays

- Parallel arrays are corresponding arrays in which values in the same relative locations are logically related

Table 5-1 Part numbers and prices for a manufacturing company

Part Number	Price
210	1.29
312	2.45
367	5.99
456	1.42

Using Parallel Arrays (continued)

```
#include<iostream>
using namespace std;
int main()
{
    const int NUMPARTS = 4;
    int partNum[NUMPARTS] = {210, 312, 367, 456};
    double partPrice[NUMPARTS] = {1.29, 2.45, 5.99, 1.42};

    int neededPart;
    int x;
    cout<<"Enter the part number you want ";
    cin>>neededPart;
    for(x = 0; x < NUMPARTS; ++x)
        if (neededPart == partNum[x])
            cout<<"The price is "<<partPrice[x]<<endl;
}
```

Figure 5-17 Program that determines part prices

```
const int NUMPARTS = 4;
int partNum[NUMPARTS] = {210, 312, 367, 456};
double partPrice[NUMPARTS] = {1.29, 2.45, 5.99, 1.42};
int isFound = 0;
int neededPart;
int x;
cout<<"Enter the part number you want ";
cin>>neededPart;
for(x = 0; x < NUMPARTS; ++x)
    if (neededPart == partNum[x])
    {
        cout<<"The price is "<<partPrice[x]<<endl;
        isFound = 1;
    }
if(isFound == 0)
    cout<<"Sorry - no such part number"<<endl;
```

Figure 5-20 Price-finding program with a variable that flags invalid part numbers

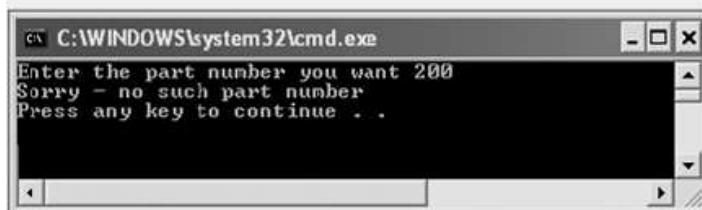


Figure 5-21 Output of the program in Figure 5-20 when user enters 200

Using Strings

- String - value expressed within double quotes
- "Hello" is a string constant
- To store a value such as 'Hello', you must create a string variable in one of two ways:
 - Create a string as an array of characters
 - Create a string using the string class defined in the C++ standard library
** preferred method in this class

Refer to the following string example available on Blackboard

```
//Program to demonstrate string input with a space
//Mrs. Alano
//

#include "stdafx.h"
#include <iostream>
#include <string>

using namespace std;

int main()
{
    string fullName;
    cout<<"Please enter your name: ";
    //cin >> fullName;    //uncomment this to see that it will only get your first name
    getline(cin, fullName);
    cout << "Your name is " << fullName << endl;

} // end of main
```

Other references

http://www.tutorialspoint.com/cplusplus/cpp_arrays.htm

<http://www.functionx.com/cpp/Lesson12.htm>