Programming Fundamentals

Video Lesson – Arrays

Name: *Griffin Gowdey*

Declare the following arrays

A) empNum, a 100-element array of ints
Ans: int empNum[100];

B) payRate, a 25-element array of doubles
Ans: double payRate[25];

C) miles, a 14-element array of longs
Ans: long miles[14];

D) letter, a 26-element array of chars
Ans: char letter[26];

E) lightYears, a 1,000-element array of doubles
Ans: double lightYears[1000];

Look at the following array definition.

int values[10];

A) How many elements does the array have?
Ans: 10

B) What is the subscript of the first element in the array?
Ans: 0

C) What is the subscript of the last element in the array?
Ans: 9

D) If an int uses four bytes of memory, how much memory does the array use?
Ans: 4 * 10 = 40

Given the following array definition:

int values [5] = { 4, 7, 6, 8, 2 } ;

index	0	1	2	3	4
value	4	7	6	8	2

What does the following statement display?

cout << values[4] << '' '' << (values[2] + values[3]) << '' '' << ++values[1] << endl;

Ans: cout << values[4] << endl; 2
cout << (values[2] + values[3]) << endl; 14
cout << ++values[1] << endl; 8

Look at the following array definition

int numbers[5] = { 1, 2, 3 };

index	0	1	2	3	4
value	1	2	3	0	0

A) What value is stored in numbers[2]?
Ans: 3

B) What value is stored in numbers[4]?
Ans: 0

Assume that array1 and array2 are both 25-element integer arrays. Indicate whether each of the following statements is legal or illegal.

A) array1 = array2;
Ans: No, because it only copies the address of the first element of array2 to array1.

B) cout << array1;
Ans: No, because it only displays the address of the first element in array1.

C) cin >> array2;
Ans: No, because we cannot enter a value in a variable that stores memory address from a keyboard.

Define four parallel arrays that hold information about 25 vehicles. Initialize the first three vehicles with the following data:

Make	Model	Year	Cost
Ford	Taurus	2006	$21,000
Honda	Accord	2004	$11,000
Jeep	Wrangler	2007	$24,000

```
Ans: string Make[25] = [“Ford, ”Honda’, ”Jeep”};
     string Model[25] = [“Taurus”, “Accord”, “Wrangler”};
     int Year[25] = {2006, 2004, 2007};
     double Cost[25] = {21000, 11000, 24000};
```

Write a loop that will step through the array you defined above, displaying the contents of each element

```
Ans: for(int k = 0; k < 25; k++
     {
          cout << setw(12) << Make[k];
          cout << setw(12) << Model[k];
          cout << setw(12) << Year[k];
          cout << setw(12) << Cost[k];
          cout << endl;
     }
```

The arrays array1 and array2 each hold 25 integer elements. Write code that copies the values in array1 to array2.

```
Ans: for(int k = 0; k < 25; k++0)
     {
          array2[k] = array1[k];
     }
```

The following code totals the values in each of two arrays described above. Will the code print the correct total for both arrays? Why or why not?

```
01:  int total = 0;
02:  //Accumulator
03:  int count;
04:  //Loop counter
05:  //Calculate and display the total of the first array.
06:  for (count = 0; count <= 25; count++)
07:       total += array1[count];
08:  cout << "The total for array1 is " << total << endl;
09:  //Calculate and display the total of the second array.
     total = 0;
10:  for (count = 0; count <= 25; count++)
11:       total += array2[count];
12:  cout << "The total for array2 is " << total << endl;
```

Ans: No, because it won't give us the single total for each individual array. It'll give us the total for both.

In a program you need to store the identification numbers of 10 employees (as ints) and their weekly gross pay (as doubles).

A) Define two arrays that may be used in parallel to store the 10 employee identification numbers and 10 weekly gross pay amounts.

```
Ans: int empIDNumbers[10];
     double weeklyGrossPay[10];
```

B) Write a loop that uses these arrays to print each employee's identification number and weekly gross pay.

```
Ans: for(int k = 0; k < 19; k++)
     {
          cout << empIDNumbers[k] << " " << weeklyGrossPay[k] << endl;
     }
```

In a program you need to store the names and populations of 12 countries. Create an appropriate array to store this information and then write the code needed to read the information into the array from a file named pop. dat.

```
Ans: string countryNames[12];
     long countryPopulations[12];
     ifstream fin;
     fin.open("pop.dat");
     int k = 0;
     while(getline(fin, countryname[k])){
          fin >> countryPopulations[k];
          k++;
          fin.ignore(100, '\n');
      }
```

A weather analysis program uses the following array to store the temperature for each hour of the day on each day of a week.

int temp[7][24];

Each row represents a day (0 = Sunday, 1 =Monday, etc.) and each column represents a time (0 =midnight, 1 = 1 a.m., ... , 12 =noon, 13 = 1 p.m., etc.).

		12am	1am	2am	3am	4am	5am	6am	7am	8am	9am	10am	11am	noon	1pm	2pm	3pm	4pm	5pm	6pm	7pm	8pm	9pm	10pm	11pm
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
S	0													40											
M	1													58											
T	2	40	42	42	42	42	38	38	41	41	45	45	45	52	52	54	60	60	60	60	55	55	55	49	49
W	3													63											
T	4													61											
F	5													50											
S	6													49											

A) Write code to find Tuesday's average temperature.

```
Ans:  //Step 1: Find the sum of row 2
            int sum = 0;
            for(int c = 0; c < 24; c++)
            {
                  sum += temp[2][c];
            }
            //Step 2: Find the mean average
            average = sum / 24.0;
```

B) Write code to find the average weekly noon temperature

```
Ans: //Step 1: Find the sum of column 12
      sum = 0;
      average = 0;
      for(int r = 0; r < 7; r++)
      {
            sum += temp[r][12];
      }
     //Step 2: Find the mean average
     average = sum / 7;
```

Write the prototype of a function, printList, that accepts an array as a parameter and displays the element of an array.

```
Ans: void printList( int myList[], int SIZE)
     {
          for(int j = 0; j < SIZE; j++)
     {
               cout << myList[j] << endl;
          }
     }
```

Write the prototype of a function, searchList, that accepts an array as a parameter and finds a value in a an array and displays the index of the element.

```
Ans: void searchList( int myList[], int SIZE, int s)
   {
          for(int j = 0; j < SIZE; j++){
               if ( s== myList[j])
                    cout << "Found at " << j << endl;
               else
                    cout << "Not found";
          }
    }
```

Write the prototype of a function, loadList, that accepts an array as a parameter and stores the values of a file into an array

```
Ans: void loadList( int myList[], int SIZE)
     {
          ifstream fin;
          fin.open("C:\\classdata\\vinData.txt");
          int j = 0;
          while(fin >> myList[j] && j < SIZE)
               j++;
          fin.close();
     }
```

Define the function, printList()

```
Ans:  printList(arrayName, sizeVarVal);
```

Define the function, searchList()

```
Ans: searchList(arrayName, sizeVarVal);
```

Define the function, loadList()

```
Ans: loadList(someArray, 3002);
```

Write the statement that calls printList()

```
Ans: printList(list, MAX);
```

Write the statement that calls searchList()

Ans: searchList(list, MAX, val);

Write the statement that calls loadList()

Ans: loadList(list, MAX);

Create a folder on your C:\ drive and name it classdata. Double-click this icon(vinData.txt) below and save it in your C:\classdata folder

vinData.txt

Double-click this icon (arraysDemo1.txt) below. Copy the code and paste in your compiler. Compile, Build, Run and Test the program. Explore and make modifications to it. Have fun!

arraysDemo1.txt