7:00Chapter 8 – one-dimensional arrays

*by GD Iyer and Jesse Black*

# Do the following lab exercises to familiarize yourself with one-dimensional arrays.

(10 points for each question).

1. Consider the following declaration:

double passwords[100];

In this declaration, identify the following:

a. The array name.

b. The array size (the number of elements)

c. The data type of each array element.

d. The range of values for the index of the array.

2. Determine whether the following array declarations are valid. If a declaration is invalid, explain why.

a. string employees[82];

b. int myArray(50);

c. int SIZE;  
 double list[SIZE];

d. const int X = 50;  
 double list[X - 60];

e. int ids[-30];

f. scores double[10];

3. When defining an array, the array dimensions must be constants, named constants, or omitted. Write C++ statements to do the following:

a. Define a named constant to be 20, then declare an array named beta to hold that many elements of type double.

b. Initialize (at declaration time) each element of the array beta to 0.

c. Set (at execution time) each element of the array beta to 0.0

d. Set the value of the seventh element of the array beta to 70.50.

e. Set the last element of beta to the sum of the first element and the second element. When referring to the last element, use the constant you defined in 3a above.

f. Use a **for** loop (using the constant you defined in 3a above) to output the value of all elements of beta which have an *index* which is a multiple of 4.

g. Output the contents of beta so that four elements are printed per line, with at least one space between elements. Use the variable j as the loop index (the loop control variable).

4. Complete the following segment of C++ code. What is stored in myList after the following C++ code executes?

|  |
| --- |
| double myList[5] = { 1.0, 2.0 };  for (int i = 2; i < 5; i++)  {  myList[i] = … \* … ; // set this element to the product of the previous two elements  } |

5. Correct the following code so that it correctly sets the value of each element of myList to the index of the element.

int myList[10];

for (int i = 1; i <= 10; i--)  
 myList[i] = [i];

6. Write C++ statements to define and initialize the following arrays.

a. Array heights with 8 elements of type double. Initialize this array to the following values:   
1.01, 2.0, 3.8, 4.0, 2.5, 6.0, 7.7, 8.0

b. Array weights of 7 elements of type int. Initialize this array to the following values:   
120, 125, 137, 40, 150, 180, 210.

c. Array mathSymbols of type char. Initialize this array to the following values:   
'+', '-', '\*', '/', '%'.

d. Array seasons of 4 elements of type string. Initialize this array to the following values:   
"fall", "winter", "spring", "summer".

7. Given the following C++ code, replace the *???* with the code so the program will do what the comments say. Then show the output of the program.

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  int beta[7] = {0, 1};  for (int i = 2; i < 7; i++)  {  *// compute the sum of the previous two entries of beta and put the result into the i-th element of beta*  *???*  }  *// this for loop will display every element of beta, each followed by a space.*  for (int i = *???*; i < *???*; i++)  cout << *???* << " ";  cout << endl;  return 0;  } |
| The output of the program above is: |

8. Suppose that you have the following function definition:

void sum(int x, int y, int & z)

{

z = x + y;

}

Consider the following declarations:

int list1[10], list2[10], list3[10];

int a, b, c;

Which of the following function calls is/are valid? What is the reason the other(s) are *not* valid?

a. sum( a, b, c );

b. sum( list1[0], list2[0], a );

c. sum( list1, list2, c );

d. for ( int i = 1; i <= 10; i++ )   
 sum( list1[i], list2[i], list3[i] );

9. When an array is passed as an actual parameter to a function, what is actually being passed?

10. Suppose that you have the following declaration:

int list[7] = {6, 10, 14, 18, 22};

If this declaration is valid, what is stored in each element of list?

*Updated 2019-11-10 by Jesse Black*