**Jeffrey Cho**

**Chapter 8 Study Guide**

1. What would be a valid range for the index of an array of size 99? What is the index of the first element? What is the index of the last element?

**The range would be from 0 to 98. The index of first element is 0, and the index of last element is 98.**

1. Write C++ statement(s) to do the following:
   1. Declare an array alpha of 50 components of type int.

**Int alpha[50];**

* 1. Initialize each component of alpha to -1.

**Int index;**

**for (index = 0; index < 50; index++)**

**alpha[index] = -1;**

* 1. Output the value of the first component of the array alpha.

**Cout << alpha[0] << endl;**

* 1. Set the value of the twenty-fifth component of the array alpha to 62.

**Alpha[24]=62;**

* 1. Set the value of the tenth component of alpha to three times the value of the fiftieth component of alpha plus 10.

**Alpha[9] = 3 \* (alpha[49] + 10);**

* 1. Use a for loop to output the value of a component of alpha if its index is a multiple of

2 or 3.

**Int index;**

**For (index=0; index < 50; index++){**

**If (index%2 == 0 || index%3 == 0){**

**Cout << Alpha[index] << endl;**

* 1. Output the value of the last component of alpha.

**Cout << alpha[49] << endl;**

* 1. Output the value of the alpha so that 15 components per line are printed.

**Int index;**

**For (index=0; index<50; index++){**

**If (index%15 ==0)**

**Cout << “\n” << alpha[index] << “ “;**

**Cout << alpha[index]<< “ “;**

* 1. Use a for loop to increment every other element (the even indexed elements).

**Int index;**

**For (index=0; index<50; index = index+2)**

**{**

**Cout << alpha[index] << endl;**

**}**

* 1. Use a for loop to create a new array, diffAlpha, whose elements are the differences between consecutive elements in alpha.

**Int diffAlpha[50];**

**Int index;**

**For (index=0, index<50; index++){**

**diffAlpha[index]=abs(alpha[index]-alpha[index+1]);**

**}**

1. Determine whether the following array declarations are valid. If a declaration is invalid, explain why.
   1. string customers[]; **invalid; array size missing**
   2. int numArray[50]; **valid**
   3. const int SIZE = 30; double list[20 - SIZE]; **invalid; size of array is negative**
   4. d. int length = 50; double list[length - 50]; **valid**
   5. e. int ids[-30]; **invalid, size of array is negative**

f. colors [30] string; **invalid; array was not declared (No type).**

1. What is array index out of bounds? Does C++ check for array indices within bounds

**If array index is less than 0 or greater than array size − 1, we say that the array index is out of bounds. C++ does not check for array indices within bounds.**

1. Determine whether the following array declarations are valid. If a declaration is valid, determine the size of the array.
   1. int list[] = {18, 13, 14, 16}; **valid; 4**
   2. int x[10] = {1, 7, 5, 3, 2, 8}; **valid; 10**
   3. double y[4] = {2.0, 5.0, 8.0, 11.0, 14.0}; **invalid**
   4. double lengths[] = {8.2, 3.9, 6.4, 5.7, 7.3}; **valid; 5**
   5. int list[7] = {12, 13, , 14, 16, , 8}; **invalid**
   6. string names[8] = {"John","Lisa", "Chris", "Katie"}; **valid; 8**

1. Consider the following function heading:

void tryMe(int x[], int size);

and the declarations:

int list[100]; int score[50]; double gpas[50];

Which of the following function calls is valid?

* 1. tryMe(list, 100); **valid**
  2. tryMe(list, 75); **invalid; The size of score is 50, so the call should be tryMe(score, 50).**
  3. tryMe(score, 100); **valid**
  4. tryMe(score, 49); **valid**
  5. tryMe(gpas, 50); **invalid; The array gpa is of type double, whereas the parameter x of tryMe is of type int, so there will be a mismatch data type error.**

1. 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 valid?

1. **sum(a, b, c); \***
2. **sum(list1[0], list2[0], a); \***
3. sum(list1, list2, c);
4. **for (int i = 1; i <= 10; i++) sum(list1[i], list2[i], list3[i]); \***

1. When an array is passed as an actual parameter to a function, what is actually being passed? In C++, as an actual parameter, can an array be passed by value?

**In C++, arrays are passed by reference only. An array cannot be passed by value.**

1. Consider the following function heading.

void modifyList(int list[], int length)

In the definition of the function modifyList, can you use a range-based for loop to process the elements of list? Justify your answer.

**No, because during compile time the formal parameter list has no first or last elements.**

1. Suppose that you have the following declarations:

int times[30][7]; int speed[15][7]; int trees[100][7]; int students[50][7];

* 1. Write the definition of the function print that can be used to output the contents of these arrays.

**void print(int function[][int col], int row)**

**{**

**Int row1, col1;**

**for (row1 = 0; row1 < row; row1++)**

**{**

**for (col1 = 0; col 1< col; col1++)**

**cout << setw(5) << function[row1][col1] << " ";**

**cout << endl;**

**}**

**}**

* 1. Write the C++ statements that calls the function print to output the contents of the arrays times, speed, trees, and students.

**Cout << print(times[][7],30) << endl;**

**Cout << print(speed[][7],15) << endl;**

**Cout << print(trees[][7],100) << endl;**

**Cout << print(students[][7],50) << endl;**

1. Assume the following declarations:

char name[21]; char yourName[21]; char studentName[31];

Mark the following statements as valid or invalid. If a statement is invalid, explain why.

* 1. cin >> name; **valid**
  2. cout << studentName; **valid**
  3. yourName[0] = '\0'; **valid**
  4. yourName = studentName; **invalid; different array sizing and incompatible types of assignment.**
  5. if (yourName == name) studentName = name; **invalid; different array sizing and incompatible types of assignment.**
  6. int x = strcmp(yourName, studentName); **invalid; strcmp was not defined nor declared.**
  7. strcpy(studentName, name); **invalid; strcpy was not defined nor declared.**
  8. for (int j = 0; j < 21; j++) cout << name[j]; **valid**

1. Sort the following list using the selection sort algorithm as discussed in this chapter. Show the list after each iteration of the outer for loop.

6, 45, 10, 25, 58, 2, 50, 30, 86

**2, 45, 10, 25, 58, 6, 50, 30, 86**

**2, 6, 10, 25, 58, 45, 50, 30, 86**

**2, 6, 10, 25, 30, 45, 50, 58, 86**