

## **Exercise no 7**

### **7.1**

**(Fill in the Blanks)** Answer each of the following:

- a) Lists and tables of values can be stored in \_\_\_\_\_ or \_\_\_\_\_.

**Answer**

arrays, vectors

- b) An array's elements are related by the fact that they have the same \_\_\_\_\_ and \_\_\_\_\_.

**Answer:**

array name, type

- c) The number used to refer to a particular element of an array is called its \_\_\_\_\_.

**Answer:**

subscript or index

- d) A(n) \_\_\_\_\_ should be used to declare the size of an array, because it eliminates magic numbers.

**Answer:**

constant variable

- e) The process of placing the elements of an array in order is called \_\_\_\_\_ the array.

**Answer:**

sorting

- f) The process of determining if an array contains a particular key value is called \_\_\_\_\_ the array.

**Answer:**

searching

- g) An array that uses two subscripts is referred to as a(n) \_\_\_\_\_ array.

**Answer:**

two-dimensional

## 7.2

(True or False) State whether the following are true or false. If the answer is false, explain why.

- a)** A given array can store many different types of values.

**Answer:**

False. An array can store only values of the same type.

- b)** An array subscript should normally be of data type float.

**Answer:**

False. An array subscript should be an integer or an integer expression.

- c)** If there are fewer initializers in an initializer list than the number of elements in the array, the remaining elements are initialized to the last value in the initializer list.

**Answer:**

False. The remaining elements are initialized to zero.

- d)** It's an error if an initializer list has more initializers than there are elements in the array.

**Answer:**

True.

## 7.3

(Write C++ Statements) Write one or more statements that perform the following tasks for an array called fractions:

- a)** Define a constant variable arraySize to represent the size of an array and initialize it to 10.

**Answer:**

```
const size_t arraySize = 10;
```

- b)** Declare an array with arraySize elements of type double, and initialize the elements to 0.

**Answer:**

```
array< double, arraySize > fractions = { 0.0 };
```

- c)** Name the fourth element of the array.

**Answer:**

```
fractions[ 3 ]
```

- d)** Refer to array element 4.

**Answer:**

```
fractions[ 4 ]
```

- e)** Assign the value 1.667 to array element 9.

**Answer:**

```
fractions[ 9 ] = 1.667;
```

- f)** Assign the value 3.333 to the seventh element of the array.

**Answer:**

```
fractions[ 6 ] = 3.333;
```

- g)** Display array elements 6 and 9 with two digits of precision to the right of the decimal point...

**Answer:**

```
cout << fixed << setprecision( 2 ); cout << fractions[ 6 ] << ' ' << fractions[ 9 ] << endl;
```

**h)** Display all the array elements using a counter-controlled for statement...

**Answer:**

```
for ( size_t i = 0; i < fractions.size(); ++i ) cout << "fractions[" << i << "] = " << fractions[ i ] << endl;
```

i) Display all the array elements separated by spaces using a range-based for statement.

**Answer:**

```
for ( double element : fractions ) cout << element << ' ';
```

## 7.4

Write C++ statements to accomplish each of the following tasks:

**a)** Task: Define a constant variable arraySize and initialize it to 10.

**Answer:**

```
const size_t arraySize = 10;
```

**b)** Task: Declare an array with arraySize elements of type double, and initialize the elements to 0.

**Answer:**

```
array< double, arraySize > fractions = { 0.0 };
```

**c)** Task: Name the fourth element of the array.

**Answer:**

```
fractions[ 3 ]
```

**d)** Task: Refer to array element 4.

**Answer:**

```
fractions[ 4 ]
```

**e)** Task: Assign the value 1.667 to array element 9.

**Answer:**

```
fractions[ 9 ] = 1.667;
```

**f)** Task: Assign the value 3.333 to the seventh element of the array.

**Answer:**

```
fractions[ 6 ] = 3.333;
```

**g)** Task: Display array elements 6 and 9 with two digits of precision.

**Answer:**

```
cout << fixed << setprecision( 2 ) << fractions[ 6 ] << " " << fractions[ 9 ] << endl;
```

**h)** Task: Display all array elements using a counter-controlled for statement.

**Answer:**

```
for ( size_t i = 0; i < fractions.size(); ++i ) cout << fractions[ i ] << " ";
```

## 7.5

**Question :** Combine the statements in Exercise 7.4 into a program.

**Answer:**

(Basically writing a main() function and including <iostream>, <array>, and <iomanip> with the code from 7.4).

## 7.6

**Fill in the blanks in each of the following:**

**a)** Question:

The names of the four elements of array p are \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_ and \_\_\_\_\_.

**Answer:**

p[0], p[1], p[2], and p[3].

**b)** Question:

Naming an array, stating its type and specifying the number of elements in the array is called \_\_\_\_\_ the array.

**Answer:**

defining.

**c)** Question: When accessing an array element, by convention, the first subscript in a two-dimensional array identifies an element's \_\_\_\_\_ and the second subscript identifies an element's \_\_\_\_\_.

**Answer:**

row, column.

**d)** Question:

An m-by-n array contains \_\_\_\_\_ rows, \_\_\_\_\_ columns and \_\_\_\_\_ elements

**Answer:**

m, n, m \times n.

**e)** Question:

The name of the element in row 3 and column 5 of array d is \_\_\_\_\_.

**Answer:**

d[2][4].

## 7.7

**(True or False) Determine whether each of the following is true or false. If false, explain why.**

**a)** To refer to a particular location or element within an array, we specify the name of the array and the value of the particular element.

**Answer:**

False. We specify the name of the array and the subscript (index) of the element.

**b)** An array definition reserves space for an array.

**Answer:**

True.

- c)** To indicate reserve 100 locations for integer array p, you write p[ 100 ].

**Answer:**

False. This would be a reference to an element, not a definition. A definition would be  
array<int, 100> p;

- d)** A for statement must be used to initialize the elements of a 15-element array to zero.

**Answer:**

False. It can be done during definition: array<int, 15> a = {0};

- e)** Nested for statements must be used to total the elements of a two-dimensional array.

**Answer:**

True. (Standard approach for row/column access).

## **7.8**

- a)** Question: Display the value of element 6 of character array alphabet.

**Answer:**

```
cout << alphabet[ 5 ] << endl;
```

- b)** Question: Input a value into element 4 of floating-point array grades.

**Answer:**

```
cin >> grades[ 3 ];
```

- c)** Question: Initialize each of the 5 elements of integer array values to 8.

**Answer:**

```
array< int, 5 > values = { 8, 8, 8, 8, 8 };
```

- d)** Question: Total and display elements of 100-element array temperatures.

**Answer:**

```
double total = 0;  
for ( double temp : temperatures )  
    total += temp;  
cout << total;
```

- e)** Question: Copy array a (11 elements) into the first portion of array b (34 elements).

**Answer:**

```
for ( size_t i = 0; i < a.size(); ++i )  
    b[ i ] = a[ i ];
```

- f)** Question: Find smallest and largest values in 99-element array w.

**Answer:**

```
double smallest = w[0];  
double largest = w[0];  
for (size_t i = 1; i < w.size(); ++i ) {  
    if (w[i] < smallest) smallest = w[i];
```

```
if (w[i] > largest) largest = w[i];
}
cout << "Smallest: " << smallest << " Largest: " << largest << endl;.
```

## **Exercise 7.9 (Two-Dimensional Array)**

Question: Consider a 2-by-3 integer array t.

a) Question: Write a declaration for t.

**Answer:**

array<array<int, 3>, 2> t;

b) Question: How many rows does t have?

**Answer:**

2 rows.

c) Question: How many columns does t have?

**Answer:**

3 columns.

d) Question: How many elements does t have?

**Answer:**

6 elements.

e) Question: Write the names of all the elements in row 1 of t.

**Answer:**

t[0][0], t[0][1], t[0][2]

f) Question: Write the names of all the elements in column 2 of t.

**Answer:**

t[0][1], t[1][1]

g) Question: Write a statement that sets the element of t in the first row and second column to zero.

**Answer:**

t[0][1] = 0;

h) Question: Write a series of statements that initializes each element of t to zero.

**Answer:**

```
for (size_t row = 0; row < t.size(); ++row)
{
    for (size_t col = 0; col < t[row].size(); ++col)
    {
        t[row][col] = 0;
    }
}
```

i) Question: Write a nested for statement that inputs the values for the elements of t from the keyboard.

**Answer:**

```

for (size_t row = 0; row < t.size(); ++row)
{
    for (size_t col = 0; col < t[row].size(); ++col)
    {
        cin >> t[row][col];
    }
}

```

## **7.10 (Salesperson Salary Ranges)**

Question: Use a one-dimensional array to solve the following problem: A company pays its salespeople on a commission basis. The salespeople receive \$200 per week plus 9% of their gross sales for that week. Write a program (using an array of counters) that determines how many of the salespeople earned salaries in each of the following ranges (assume that each salesperson's salary is truncated to an integer amount):

\$200–299

\$300–399

... (and so on up to \$1000 and over)

Answer (Logic and Code):

```

#include <iostream>
#include <array>
using namespace std;

int main() {
    array<int, 9> counters = {0}; // 9 ranges for salary
    double sales;

    cout << "Enter gross sales (-1 to end): ";
    cin >> sales;

    while (sales != -1) {
        double salary = 200 + (0.09 * sales);
        int index = static_cast<int>(salary / 100) - 2;

        if (index > 8) index = 8; // For $1000 and over
        if (index >= 0) ++counters[index];

        cout << "Enter gross sales (-1 to end): ";
        cin >> sales;
    }

    // Output formatting to show ranges
    for (size_t i = 0; i < counters.size(); ++i) {
        if (i == 8) cout << "$1000 and over: ";
        else cout << "$" << (i + 2) << "00-" << (i + 2) << "99: ";
    }
}

```

```
    cout << counters[i] << endl;
}
}
```

## **7.12**

**(Find the Errors)** Find the error(s) in each of the following statements:

**a)** Assume that a is an array of three ints. cout << a[ 1 ] << " " << a[ 2 ] << " " << a[ 3 ] << endl;

**Answer:**

Error. Referencing a[3] is an out-of-bounds error for an array of size 3 (valid indices are 0, 1, 2).

## **7.18 — Craps Game**

**Q:** How many games are won on the 1st, 2nd, ... 20th roll and after the 20th roll?

**A:** The program counts wins on each roll from 1 to 20 and after the 20th roll.

**Q:** How many games are lost on the 1st, 2nd, ... 20th roll and after the 20th roll?

**A:** The program records losses occurring on each roll from 1 to 20 and beyond.

**Q:** What are the chances of winning craps?

**A:** The chances of winning are approximately 49%.

**Q:** What does it mean that craps is a fair game?

**A:** It means the player and casino have nearly equal chances.

**Q:** What is the average length of a game?

**A:** Average length equals total rolls divided by 1000 games.

**Q:** Do chances of winning improve with longer games?

**A:** No, chances remain the same regardless of game length.

## **7.19 — Converting Vector Example to Array**

**Q:** What is changed in the program?

**A:** Vectors are replaced with arrays.

**Q:** Which features are removed?

**A:** Vector-only features like dynamic resizing are removed.

**Q:** How are elements accessed?

**A:** Elements are accessed using array indexing.

## **7.20 — What Does This Code Do?**

**Q:** What data structure is used?

**A:** A fixed-size array of 10 integers.

**Q:** What does someFunction use?

**A:** Recursion.

**Q:** What is the base condition?

**A:** When current equals array size.

**Q:** In which order are elements printed?

**A:** In reverse order.

**Q:** What is the output of the program?

**A:** 10 9 8 7 6 5 4 3 2 1

Recursive Function (What Is This)

**Q:** What does the function do?

**A:** It returns the sum of array elements.

**Q:** What is the base case?

**A:** When size equals 1.

**Q:** What is the recursive step?

**A:** Adds the last element to the sum of remaining elements.

---

---