

Homework #1

Data Structure (501324-3)

2022-2023 1st Trimester

Due: Monday September 12, 2022, 11:59 pm via Blackboard

Code 1: Initializing an array

```
2  // Initializing an array.
3  #include <iostream>
4
8  #include <iomanip>
9
10 using std::setw;
11
12 int main()
13 {
14     int n[ 10 ]; // n is an array of 10 integers
15
16     // initialize elements of array n to 0
17     for ( int i = 0; i < 10; i++ )
18         n[ i ] = 0; // set element at location i to 0
19
20     // output contents of array n in tabular format
23     for ( int j = 0; j < 10; j++ )
24         cout << j << n[ j ] << endl;
25     return 0; // indicates successful termination
27
28 } // end main
```

Code 2: Initializing an array with a declaration.

```
2  // Initializing an array with a declaration.
3  #include <iostream>
4
5
11
12 int main()
13 {
14     // use initializer list to initialize array n
15     int n[ 10 ] = { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37 };
16
17     cout << "Element" << setw( 13 ) << "Value" << endl;
18
19     // output contents of array n in tabular format
20     for ( int i = 0; i < 10; i++ )
21         cout << i << n[ i ] << endl;
22
23     return 0; // indicates successful termination
24
25 } // end main
```

Code 3: Initialize array s to the even integers from 2 to 20

```
2  // Initialize array s to the even integers from 2 to 20.
3  #include <iostream>
4
5  using std::cout;
6  using std::endl;
7
8  #include <iomanip>
9
10 using std::setw;
11
12 int main()
13 {
14     // constant variable can be used to specify array size
15     const int arraySize = 10;
16
17     int s[ arraySize ]; // array s has 10 elements
18
19     for ( int i = 0; i < arraySize; i++ ) // set the values
20         s[ i ] = 2 * i;
21
22     cout << "Element" << setw( 13 ) << "Value" << endl;
23
24     // output contents of array s in tabular format
25     for ( int j = 0; j < arraySize; j++ )
26         cout << setw( 7 ) << j << setw( 13 ) << s[ j ] << endl;
27
28     return 0; // indicates successful termination
29
30 } // end main
```

Code 4: Compute the sum of the elements of the array

```
2  // Compute the sum of the elements of the array.
3  #include <iostream>
4
5  int main()
9  {
10     const int arraySize = 10;
11
12     int a[ arraySize ] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
13
14     int total = 0;
15
16     // sum contents of array a
17     for ( int i = 0; i < arraySize; i++ )
18         total += a[ i ];
19
20     cout << "Total of array element values is " << total << endl;
21
22     return 0; // indicates successful termination
23
24 } // end main
```

Code 5: Treating character arrays as strings

```
2  // Treating character arrays as strings.
3  #include <iostream>
4
5  using std::cout;
6  using std::cin;
7  using std::endl;
8
9  int main()
10 {
11     char string1[ 20 ];           // reserves 20 characters
12     char string2[] = "string literal"; // reserves 15 characters
13
14     // read string from user into array string2
15     cout << "Enter the string \" hello there \": ";
16     cin >> string1; // reads "hello" [space terminates input]
17
18     // output strings
19     cout << "string1 is: " << string1
20         << "\n string2 is: " << string2;
21
22     cout << "\n string1 with spaces between characters is:\n";
23
24     // output characters until null character is reached
25     for ( int i = 0; string1[ i ] != '\0'; i++ )
26         cout << string1[ i ] << ' ';
27
28     cin >> string1; // reads "there"
29     cout << "\n string1 is: " << string1 << endl;
30
31     return 0; // indicates successful termination
32
33 } // end main
```

Code 6: Double-subscripted array example

```
2  // Double-subscripted array example.
3  #include <iostream>
4
5  using std::cout;
6  using std::endl;
7  using std::fixed;
8  using std::left;
9
10 #include <iomanip>
11
12 using std::setw;
13 using std::setprecision;
14
15 const int students = 3;    // number of students
16 const int exams = 4;      // number of exams
17
18 // function prototypes
19 int minimum( int [][] exams, int, int );
20 int maximum( int [][] exams, int, int );
21 double average( int [], int );
22 void printArray( int [][] exams, int, int );
23
24 int main()
25 {
26     // initialize student grades for three students (rows)
27     int studentGrades[ students ][ exams ] =
28         { { 77, 68, 86, 73 },
29           { 96, 87, 89, 78 },
30           { 70, 90, 86, 81 } };
31
32     // output array studentGrades
33     cout << "The array is:\n";
34     printArray( studentGrades, students, exams );
```

```

35
36 // determine smallest and largest grade values
37 cout << "\n\nLowest grade: "
38     << minimum( studentGrades, students, exams )
39     << "\nHighest grade: "
40     << maximum( studentGrades, students, exams ) << '\n';
41
42 cout << fixed << setprecision( 2 );
43
44 // calculate average grade for each student
45 for ( int person = 0; person < students; person++ )
46     cout << "The average grade for student " << person
47         << " is "
48         << average( studentGrades[ person ], exams )
49         << endl;
50
51 return 0; // indicates successful termination
52
53 } // end main
54
55 // find minimum grade
56 int minimum( int grades[][ exams ], int pupils, int tests )
57 {
58     int lowGrade = 100; // initialize to highest possible grade
59
60     for ( int i = 0; i < pupils; i++ )
61
62         for ( int j = 0; j < tests; j++ )
63
64             if ( grades[ i ][ j ] < lowGrade )
65                 lowGrade = grades[ i ][ j ];
66
67     return lowGrade;
68
69 } // end function minimum

```

```
70
71 // find maximum grade
72 int maximum( int grades[][ exams ], int pupils, int tests )
73 {
74     int highGrade = 0; // initialize to lowest possible grade
75
76     for ( int i = 0; i < pupils; i++ )
77
78         for ( int j = 0; j < tests; j++ )
79
80             if ( grades[ i ][ j ] > highGrade )
81                 highGrade = grades[ i ][ j ];
82
83     return highGrade;
84
85 } // end function maximum
86
87 // determine average grade for particular student
88 double average( int setOfGrades[], int tests )
89 {
90     int total = 0;
91
92     // total all grades for one student
93     for ( int i = 0; i < tests; i++ )
94         total += setOfGrades[ i ];
95
96     return static_cast< double >( total ) / tests; // average
97
98 } // end function maximum
```



```
99
100 // Print the array
101 void printArray( int grades[][ exams ], int pupils, int tests )
102 {
103     // set left justification and output column heads
104     cout << left << "                [0]  [1]  [2]  [3]";
105
106     // output grades in tabular format
107     for ( int i = 0; i < pupils; i++ ) {
108
109         // output label for row
110         cout << "\nstudentGrades[" << i << "]" << " ";
111
112         // output one grades for one student
113         for ( int j = 0; j < tests; j++ )
114             cout << setw( 5 ) << grades[ i ][ j ];
115
116     } // end outer for
117
```