

COSC 120 Lab 7 Report
Charles Reigle

Lab 7.1

Source Code

Exercise 1:

```
// Charles Reigle

#include <iostream>
using namespace std;

typedef int GradeType[100];    // declares a new data type:
// an integer array of 100 elements

float findAverage(const GradeType, int);    // finds average of all grades
int findHighest(const GradeType, int);    // finds highest of all grades
int findLowest(const GradeType, int);    // finds lowest of all grades

int main()
{
    GradeType grades;    // the array holding the grades.
    int numberOfGrades;    // the number of grades read.
    int pos;    // index to the array.
    float avgOfGrades;    // contains the average of the grades.
    int highestGrade;    // contains the highest grade.
    int lowestGrade;    // contains the lowest grade.

    // Read in the values into the array
    pos = 0;
    cout << "Please input a grade from 1 to 100, (or -99 to stop)" << endl;
    cin >> grades[pos];

    int curr;
    while (true)
    {
        cin >> curr;
        if (curr == -99) { break; }

        pos++;
        grades[pos] = curr;
    }

    numberOfGrades = pos + 1;

    avgOfGrades = findAverage(grades, numberOfGrades);
    cout << endl << "The average of all the grades is " << avgOfGrades << endl;

    highestGrade = findHighest(grades, numberOfGrades);
    cout << endl << "The highest grade is " << highestGrade << endl;

    lowestGrade = findLowest(grades, numberOfGrades);
    cout << endl << "The lowest grade is " << lowestGrade << endl;

    return 0;
}
```

```
float findAverage(const GradeType array, int size)
{
    float sum = 0;           // holds the sum of all the numbers

    for (int pos = 0; pos < size; pos++)
        sum = sum + array[pos];

    return (sum / size);     // returns the average
}
```

+ // ...

```
int findHighest(const GradeType array, int size)
{
    int highest = array[0];

    for (int i = 1; i < size; i++) {
        if (array[i] > highest) highest = array[i];
    }

    return highest;
}
```

+ // ...

```
int findLowest(const GradeType array, int size)
{
    int lowest = array[0];

    for(int i = 1; i < size; i++) {
        if (array[i] < lowest) lowest = array[i];
    }
    return lowest;
}
```

Exercise 3:

```
int main()
{
    GradeType grades; // the array holding the grades.
    ifstream dataIn;
    int numberOfGrades; // the number of grades read.
    int pos; // index to the array.
    float avgOfGrades; // contains the average of the grades.
    int highestGrade; // contains the highest grade.
    int lowestGrade; // contains the lowest grade.

    // Read in the values into the array
    dataIn.open("gradfile.txt");
    pos = 0;

    while (!dataIn.eof()) {
        dataIn >> grades[pos];
        pos++;
    }

    numberOfGrades = pos;

    for (int i = 0; i < numberOfGrades; i++) {
        cout << grades[i] << endl;
    }

    avgOfGrades = findAverage(grades, numberOfGrades);
    cout << endl << "The average of all the grades is " << avgOfGrades << endl;

    highestGrade = findHighest(grades, numberOfGrades);
    cout << endl << "The highest grade is " << highestGrade << endl;

    lowestGrade = findLowest(grades, numberOfGrades);
    cout << endl << "The lowest grade is " << lowestGrade << endl;

    return 0;
}
```

Output

Exercise 2:

```
Please input a grade from 1 to 100, (or -99 to stop)
```

```
90
```

```
45
```

```
72
```

```
62
```

```
-99
```

```
The average of all the grades is 67.25
```

```
The highest grade is 90
```

```
The lowest grade is 45
```

Exercise 3:

```
90
```

```
45
```

```
73
```

```
21
```

```
62
```

```
The average of all the grades is 58.2
```

```
The highest grade is 90
```

```
The lowest grade is 21
```

Lab 7.2

Source Code

```
int main()
{
    StringType30 firstname, lastname; // two arrays of characters defined
    int numOfGrades; // holds the number of grades
    GradeType grades; // grades defined as a one dimensional array
    float average; // holds the average of a student's grade
    char moreInput; // determines if there is more input

    cout << setprecision(2) << fixed << showpoint;

    // Input the number of grades for each student
    cout << "Please input the number of grades each student will receive." << endl
        << "This must be a number between 1 and " << MAXGRADE << " inclusive"
        << endl;
    cin >> numOfGrades;

    while (numOfGrades > MAXGRADE || numOfGrades < 1)
    {
        cout << "Please input the number of grades for each student." << endl
            << "This must be a number between 1 and " << MAXGRADE
            << " inclusive\n";
        cin >> numOfGrades;
    }

    // Input names and grades for each student
    cout << "Please input a y if you want to input more students"
        << " any other character will stop the input" << endl;
    cin >> moreInput;

    while (moreInput == 'y' || moreInput == 'Y')
    {
        cout << "Please input the first name of the student" << endl;
        cin >> firstname;

        cout << endl << "Please input the last name of the student" << endl;
        cin >> lastname;

        for (int count = 0; count < numOfGrades; count++)
        {
            cout << endl << "Please input a grade" << endl;

            cin >> grades[count];
        }

        cout << firstname << " " << lastname << " has an average of ";

        float average = findGradeAvg(grades, numOfGrades);
        cout << average << " which gives a letter grade of " << findLetterGrade(average);

        cout << endl << endl << endl;

        cout << "Please input a y if you want to input more students"
            << " any other character will stop the input" << endl;
        cin >> moreInput;
    }

    return 0;
}
```

Output

```
Please input the number of grades each student will receive.
This must be a number between 1 and 25 inclusive
3
Please input a y if you want to input more students any other character will stop the input
y
Please input the first name of the student
Mary
Please input the last name of the student
Brown
Please input a grade
100
Please input a grade
90
Please input a grade
90
Mary Brown has an average of 93.33 which gives a letter grade of A

Please input a y if you want to input more students any other character will stop the input
y
Please input the first name of the student
George
Please input the last name of the student
Smith
Please input a grade
90
Please input a grade
30
Please input a grade
50
George Smith has an average of 56.67 which gives a letter grade of F
```

```
Please input a y if you want to input more students any other character will stop the inputy
Please input the first name of the student
Dale
```

```
Please input the last name of the student
Barnes
```

```
Please input a grade
80
```

```
Please input a grade
78
```

```
Please input a grade
82
```

```
Dale Barnes has an average of 80.00 which gives a letter grade of B
```

```
Please input a y if you want to input more students any other character will stop the inputy
Please input the first name of the student
Sally
```

```
Please input the last name of the student
Dolittle
```

```
Please input a grade
70
```

```
Please input a grade
65
```

```
Please input a grade
80
```

```
Sally Dolittle has an average of 71.67 which gives a letter grade of C
```

```
Please input a y if you want to input more students any other character will stop the inputy
Please input the first name of the student
Conrad
```

```
Please input the last name of the student
Bailer
```

```
Please input a grade
60
```

```
Please input a grade
58
```

```
Please input a grade
71
```

```
Conrad Bailer has an average of 63.00 which gives a letter grade of D
```

Lab 7.3

Source Code

Exercise 1:

```
19 int main()
20 {
21     int rowsUsed;           // holds the number of rows used
22     int colsUsed;           // holds the number of columns used
23     PriceType priceTable;   // a 2D array holding the prices
24
25     getPrices(priceTable, rowsUsed, colsUsed); // calls getPrices to fill the array
26     printPrices(priceTable, rowsUsed, colsUsed); // calls printPrices to display array
27
28     return 0;
29 }
30
31 // ...
32
42
43 void getPrices(PriceType table, int& numOfRows, int& numOfCols)
44 {
45     cout << "Please input the number of rows from 1 to " << MAXROWS << endl;
46     cin >> numOfRows;
47
48     cout << "Please input the number of columns from 1 to " << MAXCOLS << endl;
49     cin >> numOfCols;
50
51     for (int row = 0; row < numOfRows; row++)
52     {
53         for (int col = 0; col < numOfCols; col++)
54         {
55             cout << "Please input the price of an item with 2 decimal places" << endl;
56             cin >> table[row][col];
57         }
58     }
59 }
60
61 // ...
62
70
71 void printPrices(PriceType table, int numOfRows, int numOfCols)
72 {
73     cout << fixed << showpoint << setprecision(2);
74
75     for (int row = 0; row < numOfRows; row++)
76     {
77         for (int col = 0; col < numOfCols; col++)
78         {
79             cout << left << setw(7) << table[row][col] << " ";
80         }
81         cout << endl;
82     }
```


Exercise 4:

```
float findLowestPrice(PriceType table, int rows, int cols) {  
    float lowest;  
  
    lowest = table[0][0];  
    for (int row = 0; row < rows; row++) {  
        for (int col = 0; col < cols; col++) {  
            if (table[row][col] < lowest)  
                lowest = table[row][col];  
        }  
    }  
    return lowest;  
}
```

Exercise 7:

```
void printSales(SalesType table, int numOfYear){  
    for (int y = 0; y < numOfYear; y++) {  
        for (int c = 0; c < MAXCOL; c++) {  
            if (c == 0) cout << setw(0) << table[y][c];  
            else if (c == 1) cout << setw(6) << table[y][c];  
            else cout << setw(10) << table[y][c];  
        }  
        cout << endl;  
    }  
}
```

Output

Exercise 1:

```
Please input the number of rows from 1 to 10
2
Please input the number of columns from 1 to 10
3
Please input the price of an item with 2 decimal places
1.45
Please input the price of an item with 2 decimal places
2.56
Please input the price of an item with 2 decimal places
12.98
Please input the price of an item with 2 decimal places
37.86
Please input the price of an item with 2 decimal places
102.34
Please input the price of an item with 2 decimal places
67.89
1.45    2.56    12.98
37.86   102.34  67.89
```

Exercise 3:

```
Please input the number of rows from 1 to 10
2
Please input the number of columns from 1 to 10
3
Please input the price of an item with 2 decimal places
1.45
Please input the price of an item with 2 decimal places
2.56
Please input the price of an item with 2 decimal places
12.98
Please input the price of an item with 2 decimal places
37.86
Please input the price of an item with 2 decimal places
102.34
Please input the price of an item with 2 decimal places
67.89
1.45    2.56    12.98
37.86   102.34  67.89
Highest Price: 102.34
```

Exercise 4:

```
Please input the number of rows from 1 to 10
2
Please input the number of columns from 1 to 10
3
Please input the price of an item with 2 decimal places
1.45
Please input the price of an item with 2 decimal places
2.56
Please input the price of an item with 2 decimal places
12.98
Please input the price of an item with 2 decimal places
37.86
Please input the price of an item with 2 decimal places
102.34
Please input the price of an item with 2 decimal places
67.89
1.45    2.56    12.98
37.86   102.34  67.89

Highest Price: 102.34

Lowest Price: 1.45
```

Exercise 7:

```
Please input the number of years (1-10)
3
Enter the number of transactions for Year 1 Quarter 1: 72
Enter the number of transactions for Year 1 Quarter 2: 80
Enter the number of transactions for Year 1 Quarter 3: 60
Enter the number of transactions for Year 1 Quarter 4: 100
Enter the number of transactions for Year 2 Quarter 1: 82
Enter the number of transactions for Year 2 Quarter 2: 90
Enter the number of transactions for Year 2 Quarter 3: 43
Enter the number of transactions for Year 2 Quarter 4: 98
Enter the number of transactions for Year 3 Quarter 1: 64
Enter the number of transactions for Year 3 Quarter 2: 78
Enter the number of transactions for Year 3 Quarter 3: 58
Enter the number of transactions for Year 3 Quarter 4: 84
    YEARLY QUARTERLY SALES

YEAR Quarter 1 Quarter 2 Quarter 3 Quarter 4
2000    72      80      60      100
2001    82      90      43      98
2002    64      78      58      84
```

Answers:

Exercise 2: getPrices passes the values by reference so that it can modify the values in the function, and then in the main function where they are declared, they can then be passed to the printPrices function. If they weren't passed by reference in getPrices, the values would never be assigned to anything in the main function.