Department of Electrical and Computer Engineering
University of Massachusetts Dartmouth

ECE160: Foundations of Computer Engineering I (Spring 2023)
Instructor: Dr. Liudong Xing

LAB # 10

(Relevant Lecture: #18, 20, 21)

Monday, April 3 (L1) and Wednesday, April 5 (L2)

OBJECTIVES

- To practice how to open a file using fopen()/fopen_s() and different modes ("r", "w", "a") (L#18)
- To practice how to read from a file using fscanf()/fscanf_s() (L#20)
- To practice how to write to a file using fprintf() (L#20)
- To practice how to use arrays (L#21)

SUBMISSION REQUIREMENT

1. Please follow "Submission Guidelines" in the lab section of the course website to submit your solution (program files) to the class M: drive by **5pm, Wednesday, April 5**
2. Suggested format for naming your solution files: lab#-your last name-p#.cpp
   For example: lab10-xing-p1.cpp for problem 1; lab10-xing-p2.cpp for problem 2; …

EXERCISES

1. Modify your program or the program in the Lab#4 solution file for **Lab#4-Exercise#3** so that the program can perform the following tasks:

- read Tom's grades (float type) for four courses from last semester from the keyboard using scanf_s(),
- compute his average GPA,
- display the computed average GPA on the screen using printf(), and
- write all the input grades and the computed average GPA to a report file named TomGrade1.txt using fprintf().

Check the file TomGrade1.txt in your folder after you run the program to confirm the content.

Test case:

If you input Tom's 4 grades as 3.7 4.0 3.3 3.7 from the keyboard, then Tom's GPA 3.675000 should be displayed on the screen, and the file TomGrade1.txt should contain the following (the part highlighted in yellow is optional but recommended):

Tom's grades are 3.700000 4.000000 3.300000 3.700000, and GPA is 3.675000.

2. The program in Exercise#1 can handle one student's grades. Modify your program in Exercise #1 so that the program can handle three students' grades and GPA calculation and reporting **using a loop**. For each student, the following tasks must be performed:

- read the student's grades for four courses from last semester from the keyboard,
- compute this student's semester GPA,
- display the computed GPA on the screen using printf(), and
- write all the input grades and the computed GPA to a report file named StudentGrade.txt using fprintf().

After you run the program, the file StudentGrade.txt in your folder should contain grades and GPA for all the three students on three different lines.

Test case:

If you input student 1's 4 grades as 3.0 4.0 3.7 3.3, student 2's 4 grades as 4.0 4.0 3.7 4.0, and student 3's 4 grades as 3.0 2.7 3.3 3.7, then the file StudentGrade.txt should contain the following (the part highlighted in yellow is optional but recommended):

Student 1's grades are 3.000000 4.000000 3.700000 3.300000, and GPA is 3.500000.
Student 2's grades are 4.000000 4.000000 3.700000 4.000000, and GPA is 3.925000.
Student 3's grades are 3.000000 2.700000 3.300000 3.700000, and GPA is 3.175000.

3. Modify your program in Exercise#1 so that the program can perform the following tasks:

- write Tom's grades for four courses from last semester in the displayed format (one per line) to a file named TomGrade2.txt using fprintf(),

  3.3
  3.0
  4.0
  3.7

- read the data from the file TomGrade2.txt using fscanf_s(),
- compute the semester GPA,
- display the four course grades and computed GPA on the screen.

4. The program below assigns values to a 1-D array and then outputs each array element.
   a) Using defined constant as the size of the array provides flexibility: change the array size to 7, and then run the program, and record and understand your output.
   b) Change < array_size to < =array_size in the test expression part of the for loop (i.e., the part highlighted in yellow), then see and record what happens.

While you are only required to submit the program files for a) to the M: drive, please understand the program and why < =array_size would cause a run time error.

```
#include "stdio.h"

#define array_size 5
#define my_const 10

void main(void)
{
	int myarray[array_size];
	int i;
```

```
        for (i = 0; i < array_size; i++)
        {
                myarray[i] = i * my_const;
                printf("myarray[%d] is: %d\n", i, myarray[i]);
        }

}
```

5. Modify the program in Exercise (4) of Lecture #21 (**Slide #7 in the solution file**) so that the program reads 10 integer numbers into a 1-D array from the keyboard, finds and prints out the **maximum** (int type), **minimum** (int type) and **average** (float type) of the 10 array elements.

   Testing: if you input 1 3 5 7 9 2 4 6 8 10 from the keyboard, then the output on the screen should be:

   The largest array member is: 10
   The smallest array member is: 1
   The average is: 5.500000