

# Lab 12 Work

You always create a separate C++ file for each program you write in the lab.

1. Write a C++ program that does the following:
  - a. Create a C++ file with the name **problem1.cpp**.
  - b. Prompt the user to enter 10 integers and store each integer into a separate element of a 10-element integer array.
  - c. Compute a, the average of the 10 integers.
  - d. Print the average a.
  - e. Compute b, the average of those array elements with value greater than or equal to a.
  - f. Print the average b.

Sample run of the program:

Enter ten numbers: 12 9 34 22 6 45 76 2 34 19

Average of the integers in array: 25.9

Average of the integers greater than or equal to the overall average of 25.9 is: 47.25

2. Write a C++ program that does the following:
  - a. Create a C++ file with the name **problem2.cpp**.
  - b. Declare a string array of size 3 and initialize it to {"[Your name]", "Alice", "Bob"} where you put your actual first name in place of the text **[Your name]**.
  - c. Declare an integer array of size 3 and initialize it to {100, 90, 80}.
  - d. Use a for loop to print the following:  
[Your name]'s score is 100.  
Alice's score is 90.  
Bob's score is 80.

3. Write a C++ program that does the following:
  - a. Create a C++ file with the name **problem3.cpp**.
  - b. Prompt the user to enter a positive integer n that is at most 100, terminating the program if the user enters an integer that is less than 1 or more than 100.
  - c. Read and store n words from the user.
  - d. Print the n words in reverse order.

Sample run of the program:

How many words (at most 100)? **5**

Enter all word(s) below:

**Freddy and Max were absent**

I will now print your word(s) in reverse order:

absent were Max and Freddy

4. Write a C++ program that does the following:
- Create a C++ file with the name **problem4.cpp**.
  - Prompt the user to enter a positive integer n that is at most 100, terminating the program if the user enters an integer that is less than 1 or more than 100.
  - Read and store n integers from the user.
  - Each of the n integers is the length of a row of stars.
  - Print the n rows of stars.

Sample run of the program:

```
How many rows (at most 100)? 4
Enter 4 row lengths: 2 7 1 5
**
*****
*
*****
```

5. Write a C++ program that does the following:
- Create a C++ file with the name **problem5.cpp**.
  - Prompt the user to enter a positive integer n that is at most 100, terminating the program if the user enters an integer that is non-positive or more than 100.
  - Read and store n integers from the user.
  - Print the negative integers from the array, in order.
  - Print the positive integers (include 0 in this group) from the array, in reverse order.

Sample run of the program:

```
Enter a positive integer (at most 100): 8
Enter 8 integers: 3 -1 4 -10 17 18 19 -11
-1 -10 -11
19 18 17 4 3
```

Lab Work Submission:

- You can continue to work on this lab after our lab class, on your own, at home.
- Submit your lab work via Blackboard on or before: **Sunday, May 11, 2025**.
- This is the only accepted submission method!
- Once you submit your assignment you will not be able to resubmit it!
- Make absolutely sure the C++ files you want to submit are the C++ files you want graded.
- You will not be able to submit your lab work under any circumstances once **Lab 12 Work** disappears at **12:00 a.m. on Monday, May 12, 2025**.
- There will be **NO** exceptions to these rules!
- To submit your lab work, upload the 5 C++ files you did for this lab (**with .cpp extension**) to the **Lab 12 Work** assignment in the **Lab Work** tab on Blackboard.
- Then, make sure you click the **Submit** button to submit your lab work.

The lab work is worth a total of **9** points based only on grading one of the problems (**a randomly chosen one**).

Grading steps for the chosen problem are as follows:

- If your program does NOT compile successfully, then the grade for the lab is zero.
- If your program produces runtime errors or does NOT produce the expected output, then the grade for the lab is zero.
- If the program compiles, runs, and produces the expected output, then the grade is computed as follows:
  - 8 points – the program compiles, runs, and produces the expected output
  - 1 point – proper indentation and formatting of the code