

## CS 162 LAB #2 – Arrays

Each lab will begin with a brief demonstration for the core concepts examined in this lab. As such, this document will not serve to tell you everything the TAs/instructor will do in the demo. It is highly encouraged that you ask questions and take notes.

In order to get credit for the lab, you need to be checked off by the end of lab. You can earn a maximum of 3 points for lab work completed outside of lab time, but you must finish the lab before the next lab. For extenuating circumstances, contact your lab TAs and the instructor.

---

This lab is worth 15 points total. Here's the breakdown:

- Part 1: Worksheet (5 pts)
  - Part 2: Understand References and Pointers (Group work: 2 pts)
  - Part 3: Array Exercise (Individual work: 5 pts)
  - Part 4: Implement arrays in assignment 1 (Individual work: 3 pts)
- 

### **(5 pts) Part 1: Worksheet**

This session will be led by your lab TAs. Please follow their instructions, participate, and complete worksheet 2:

<https://classes.engr.oregonstate.edu/eecs/winter2024/cs162-001/labs/WS2.docx> (pdf version)

### **Get the start code**

Upload/download the start code for this lab onto `flip.engr.oregonstate.edu`:  
wget <https://classes.engr.oregonstate.edu/eecs/winter2024/cs162-001/labs/lab2.zip>

**Note: if are new to Linux commands, please refer to the following cheat sheet:**

[Linux Command Reference](#)

### **(2 pts) Part 2: Understand References and Pointers**

In part 2, you will be working **in a group** to review the references in C++, and we will introduce the concept of pointers.

Files needed for this part: `sentence.cpp`

#### **Revisit references:**

Suppose our main function is as follows:

```
int main() {
    string sentence;

    get_sentence(sentence);
    cout << sentence << endl;

    return 0;
}
```

In last lab, we know that in order to change the value of the string `sentence` inside the `get_sentence()` function, we need to add an ampersand (&) in front of the parameter. This is called pass by reference, i.e., by using the following function:

```
void get_sentence(string &s);
```

The reference variable `s` in the parameter list refers to the memory location of the argument, `sentence`. Therefore, if we update `s` in `get_sentence()` function, `sentence` in `main()` will be changed as well. This is accomplished by using a very important concept implicitly, pointers.

### Introduction to Pointers

In C/C++, pointers are powerful entities that allow you to manage memory and manipulate data directly.

**A pointer is a variable that holds the memory address of another variable.**

To **declare a pointer**, use the data type followed by an asterisk (\*) and the pointer name. For example:

```
int number = 42;
int *ptr = &number; //Declare and Initialize ptr with the address of 'number'
```

Now, our pointer variable, `ptr`, holds the memory address of `number`.

(Note, you may use `&var` to get the memory address of `var`)

To **access/update** the value stored at the memory address pointed to by a pointer, use the dereference operator (\*). For example:

```
cout << *ptr << endl; // the value of 'number' will be printed, which is 42
*ptr = 50; // 'number' is changed to 50
```

Since reference variables is only available in C++, you are still required to understand how to use pointers to achieve the same goal. Now, change the function prototype to:

```
void get_sentence(string *s);
```

Answer the following questions:

1. Can the value of the string (`sentence`) be changed with the modified function prototype? If so, how?
2. Compared to using reference variables, what other modifications are required when making the function call and inside the function?
3. What is the difference between an ampersand (`&`) and an asterisk (`*`) added in front of the parameter? Use a diagram to explain.

### **(5 pts) Part 3: Array Exercise**

In part 3, you will be working **individually** to finish the following exercise using 1D static array.

Files needed for this part: `array.cpp`

1. Prompt the user for the size of the array, ensuring the size is between 5 and 50, inclusive. If an invalid size is given, re-prompt until a valid size is provided. (Note: you may assume that the user will enter an integer value. (i.e., you are required to handle errors of the same data type))
2. Implement a function named `populate_array()` that takes an integer array, and the array size as parameters and fills the array with randomly generated values. Each element in the array should be a random integer between 1 and 100, inclusive.

```
void populate_array (int array[], int size);
```

3. Print out the following statistics of the generated array:
  - o The sum of all elements in the array.

- The maximum and minimum value among the array elements.
  - The average of all array elements.
4. After presenting the statistics, ask the user if they want to perform the entire process again. If the user chooses to repeat, go back to step 1. If they choose to quit, end the program.

### **(3 pts) Part 4: Implementing arrays in assignment 1**

Task one: Create an array to keep track of the shooting result of the player.

- What is the dimension and size of the array?
- How to display the result of each rack using “X”, “O”, and “M”?
- Could you reuse the array to store the shooting result of the other player?

Task two: How would you calculate the total score for all players?

- What are the steps to calculate the total score for one player?
- Do you need a separate array to store the total score for all players?

**If any of your functions are more than 15 lines of code, what can you do to make it smaller? If you are having difficulty thinking of how to make it smaller, then ask a TA to help you!!!**

Submit your work to TEACH for our records (**Note: you will not get points if you don't get checked off with a TA!!!**)

1. Transfer all files you've created in this lab from the ENGR server to your local laptop.
2. Go to [TEACH](#).
3. In the menu on the right side, go to **Class Tools** → **Submit Assignment**.
4. Select **CS162 Lab2** from the list of assignments and click “**SUBMIT NOW**”
5. Select your files and click the Submit button.

**Show your completed work and answers to the TAs for credit. You will not get points if you do not get checked off!**