CS 161 Lab #5 – Passing Arguments to Functions

- Start by getting checked off for (up to) 3 points of work from Lab 4 in the first 10 minutes.
- Your goal is to get all points for Lab 5 checked off by a TA by the end of this lab, so that you do not need to do additional work outside of the lab.
- You will work in small groups to complete the programming assignments, with each person writing their own program. Include each partner's name in your file header.

Goals:

- Practice defining functions that use reference parameters
- Practice passing arguments by value and by reference
- Practice passing arrays to functions and changing their values

---

(3 pts) A. Practice Passing Function Arguments by Value

Create a file called `lab5.cpp` with a `main()` function (and a file header with all partners' names, the lab, and the date).

✓ 1. Above `main()`, define a function as follows:

```
int plus_10_value(int n) {
   ...
}
```

that returns the value `n + 10`. Inside `main()`, read in an integer from the user and use your function to update this number by adding 10 to it (calling your function with their input and assigning the result back to the same variable). For example:

```
int main() {
   int number;
   cout << "Enter an integer: ";
   cin >> number;
   number = plus_10_value(number);
   cout << "Your number plus 10 is " << number << endl;
   return 0;
}
```

2. Add a complete function header above your function, with 3 test cases.

```
/*********************************************************************
 ** Function: plus_10_value
 ** Description:
 ** Parameters:
 ** Test cases:
 *********************************************************************/
```

Ask for a TA to check your work and give you points for this part. Start on part B while waiting.

?

(4 pts) B. Practice Passing Function Arguments by Reference

1. Now write a second function that accomplishes the same thing, but instead of returning the new value, it modifies the input parameter:

```
void plus_10_ref(int& n) {
  ...
}
```

Note the `void` return type and the & next to the input parameter `n` (`int&`), indicating that `n` is **passed by reference** (and can be modified inside the function). This function should update `n` directly instead of returning the new value.

2. Update your `main()` function to read a second value, `number2,` from the user and call

```
   plus_10_ref(number2);
```

then print `number2` afterwards.

Notice that there is no assignment because there is no value returned from this function.

Run your program a few times with appropriate test cases to ensure that both functions work correctly.

3. Ask yourself (discuss in your group or with a TA if you are stuck):

- If you remove the & from the parameter list for the **definition** of `plus_10_ref()`, what happens?
- If you add a & to the **call** to `plus_10_ref(&number2);` what happens?

(2 pts) C. Practice Passing Arrays to Functions

1. Update your `main()` function to declare an array of integers `array1` and initialize it with the numbers 1,2,3,4,5 (5 values). Add a comment to note that the sum of these values is 15.

2. Create two more arrays `array2` and `array3` with **values that you choose (test cases**) in `main()` and use a comment to note what each array's sum should be.

3. Design a `tally` function that takes in an integer array and the size of the array (two parameters) and returns the sum of all values in the array. Your function should have this form:

```
int tally(int array[], int n_items) {
    ...
}
```

Design considerations:

- What kind of loop will you use to iterate over the array values and compute the sum?
- Why does this function need `n_items` as a parameter?
- What variable(s) will you need to declare inside the `tally()` function to store the sum?
- How will you return the computed sum to the caller?

3. Implement your function.

4. Call your function on the array in your `main()` function and print the sum of the array values.

```
int s = tally(array1, 5);
cout << " The sum of array1 is " << s << endl;
```

5. Ask yourself (discuss in your group or with a TA if you are stuck):

- Did you get 15?
- What happens if you incorrectly specify the number of items?

```
int s = tally(my_array, 4);
int s = tally(my_array, 6);
```

- Try your other two test cases as well (`array2` and `array3`).

(1 pt) D. Practice Changing Array Values Inside a Function

1. Update your `main()` function to declare an array of integers called `array4` , and initialize it with the numbers 9,19,29,39,49 (5 values), and print all values in the array with a for loop:

```
   for (int i = 0; i < 5; i++) {
      cout << array4[i] << " ";
   }
   cout << endl;
```

2. Implement an `inc_each_element_by_one()` function that takes in an integer array and the size of the array (two parameters) and increments each value in the array by 1. Your function should have this form:

```
void inc_each_element_by_one(int array[], int n_items) {
   for (int i = ...... ) {
      array[i] = ...
          .
          .
          .
     }
}
```

3. Call your function on `array4` in your `main()` function, then print array values again.

```
   inc_each_element_by_one(array4, 5);
```

Did you get 10 20 30 40 50?

E. Get your work checked off by a TA.

Submit your program on Canvas (can be done while you are waiting to be checked off).

All partners' names must be in the comment header to get credit.

1. Transfer your .cpp file from the ENGR servers to your local laptop.
2. Go to the Lab 5 assignment:
   https://canvas.oregonstate.edu/courses/1770357/assignments/7847927
3. Click "Submit Assignment".
4. Upload your .cpp file (lab5.cpp).
5. Click the **"Submit Assignment"** button.
6. You are done!

If you finish the lab early, this is a golden chance to work on your Assignment 3 program (with TAs nearby to answer questions!).

Remember, the assignment you submit must be your work alone (no partners).
