**Lab Evaluation 2**

Lab Objectives:

* To be familiar with the syntax of function definition/calling and its variations
  + Function prototypes
  + Void function and formal/actual arguments
  + Typed functions and formal/actual arguments

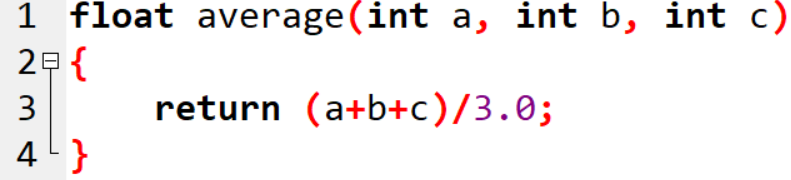
**Lab Task 1**

[Budgeted time: 15 minutes] Write down the following function prototypes only in your program. Take any name of the function, e.g. testFunc.

1. Input: nothing, Output: nothing
2. Input: an integer, Ouput: nothing
3. Input: 3 integers, Ouput: nothing
4. Input: 3 integers, Ouput: integer
5. Input: an integer and a float, Ouput: float

**Lab Task 2**

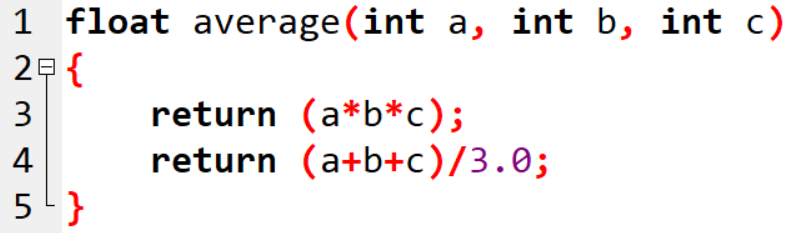
[Budgeted time: 25 minutes] Write down the following function definition in your program after the main function. Also define its prototype before the main function. (Note 3.0 instead of 3 at line 3. Why?)



1. Call it in main for with 3 integer ***constants*** 5, 6 and 8 as actual arguments, and save the result in a variable named avg. Display avg in main to verify that the result is correct.
2. Call it in main as the following statement: average(3, 4, 5); // Don’t save the result

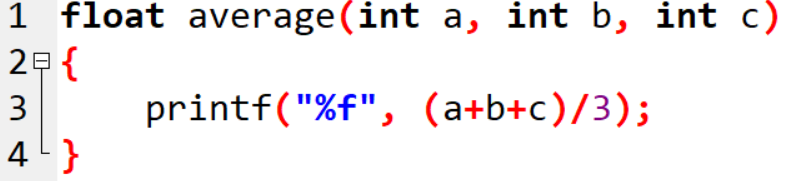
Is there any output? Why?

1. Try the following function calls. In each case, Does this statement compile successfully. If yes, then what is the output; if no, then why and remember the error for future reference?
   1. average(2, 6, average(5, 10, 15));
   2. average(2, 6, average(sqrt(25), 10, 15));
   3. cout<< average(2, 7, 1);
2. Change the above function as follows.



Now, call it in main for with 3 integer ***constants*** 5, 6 and 8 as actual arguments, and save the result in a variable named *avg*. What is the value of *avg* and why?

1. Now, change the function as follows (no return statement).



Now, call it in main for with 3 integer ***constants*** 5, 6 and 8 as actual arguments, and save the result in a variable named *avg*. Print the value of avg **inside main**. What is the value of *avg* and why?

**Lab Task 3**

[Budgeted time: 15 minutes]

1. Write a function which receives 2 Input characters as formal parameters, and returns the larger character in alphabetical order (Hint: the characters are compared on the basis of their ascii values).
2. Call this function inside main and verify for different characters as actual arguments.

**Practice Task 1.**

Write down a program, which applies Newton’s 1st, 2nd and 3rd laws of motion.

Make separate function for each law, i.e. first\_law(), second\_law(), third\_law(). Identify input(s) and output of each function and implement the functions accordingly. Identify the function parameters and the value returned by the function yourself.

Test your functions by calling them with appropriate data in main() and display the output of these functions in main(). Verify that functions are providing correct output.