

# CS300: Programming Languages

Spring 2022

## Lab Assignment #3

- Name only: \_\_\_\_\_
- Release date: Mar 08, 2022 (Tuesday), 12:30pm
- Due date: Mar 22, 2022 (Tuesday), 12:30pm
- Assignment should be SUBMITTED on Blackboard before Due Date. Other submission methods will NOT be accepted.
- LATE Submission will NOT Be Accepted on Blackboard since the submission link will be closed automatically after due date;
  - Additional submission for missing answer will NOT Be Accepted.
- There will be NO re-submission if the student submitted the wrong assignment. So, please verify the uploaded assignment after the submission.
- It should be done INDIVIDUALLY;
- Total: 10 pts

1. Write a C program to print the following table. **struct** is required to be used in the program. [5 pts]

- Before writing the program, you need to think about the following:
  - One structure only can store one record (or one row) in the table. But, there are 11 records (or 11 rows). Do you want to create 11 structures (one for each record) in the program? Or do you want to create an array of structures?
- The following references will help you solve this lab question.
  - [https://www.tutorialspoint.com/cprogramming/c\\_structures.htm](https://www.tutorialspoint.com/cprogramming/c_structures.htm)
  - <https://www.tutorialspoint.com/explain-the-array-of-structures-in-c-language>
  - <https://www.delftstack.com/howto/c/printf-align-columns-in-c/>

EXERCISE	MUSCLES	WEIGHT	TIME	SETS	REPS
1. bench_press	chest_tri_shoulders	0	0	0	0
2. squat	gluts_hamstrings_quads_calves	0	0	0	0
3. dead_lift	hamstrings_lowerBack	0	0	0	0
4. power_clean	total_body	0	0	0	0
5. hip_thrust	hamstrings_gluts	0	0	0	0
6. russian_twists	abs_obliques	0	0	0	0
7. leg_raises	lower_abs	0	0	0	0
8. bicep_curls	biceps	0	0	0	0
9. tricep_pulldowns	triceps	0	0	0	0
10. tricep_kickbacks	triceps	0	0	0	0
11. situps/crunches	abs	0	0	0	0

2. Printing Memory Addresses [5 pts]

- a. Write a short C program that declares and initializes (to any value you like) a double, an int, and a string. Your program should then print the address of, and value stored in, each of the variables. Use the format string "%u" to print the addresses as unsigned (32-bit non-negative) integers. [2 pts]

Hint: Remember that you can use the & character to find addresses.

Reminder: 1 byte = 8 bits, and a 32-bit integer requires the space of 4 bytes.

- Reference: [https://www.tutorialspoint.com/cprogramming/c\\_pointers.htm](https://www.tutorialspoint.com/cprogramming/c_pointers.htm)

- b. Draw a small memory diagram showing the location of each of the variables in your program. Are they allocated in the same order that you declared them? Is there any empty space between them? [1 pt]
- c. Modify your program by rearranging the variable declarations and/or changing the length of the string. (In particular, try a string that uses 5 or 7 bytes, including the null terminator.) Does this change the results you got previously? [2 pts]

### **Questions for the Lab**

Your submission should include the following:

- Screenshot of running results and explanation of results.
- Source codes