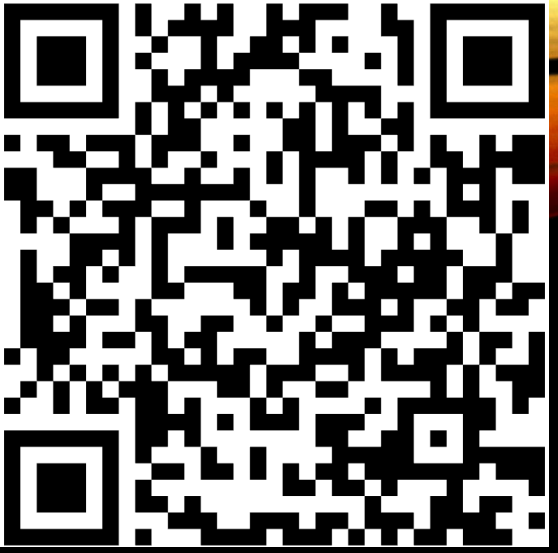


Lab GH Link: https://github.com/swiftlydesigner/122-Practice-Review

CPTS 122 L3

Boring Part – Lab Guidelines

REVIEW CLASS EXPECTATIONS

- Lab Canvas Page > Modules > Week 1 > Expectations.pdf

ASKING FRIENDS/COLLEAGUES FOR HELP

- Set aside time for them to **explain** how to complete the objective
- DO NOT COPY/PASTE THEN EXPLAIN – THIS IS PLAGARISM
- While you can work with others on higher-level features (UI/UX), credit must be given
  - Please keep this code to a minimum. If there is too much similar work (even credited), it will likely end in with 50+ points off
- Be sure that everybody involved can write the code for the objective

OFFICE HOURS

- Dana 134
- Day: TBA
- Hours: TAB
  - I can show up early upon 18-hour notice (TBA)
- Email for in-person or Zoom appointment if you cannot make it or need more help. Depending on my schedule, I may do Zoom appointments on the weekends and over breaks.

CONTACT

- **(Preferred)** Message me on MS Teams @ kyle.parker
- Email me at kyle.parker@wsu.edu
- I typically respond within an hour if I am not in class or working on an assignment. Give me 24-48 hours incase I am busy some day
- I can only respond to emails from your WSU email (FERPA regulation)
- If you email me, **Include "CPTS 122 L3:" in the subject**
- Contact me to:
  - Setup appointments
  - Ask questions or need clarifications on assignments
  - Let me know you are missing a lab
  - Let me know you are missing an exam (Attempt to contact Andy first)
    - Limited acceptable reasons

GRADING

- Expect exams back the next day, sometimes two
- Expect quizzes back on Thursdays
- Expect PAs back within a week (possibly up to 2 weeks later in the semester)
  - You will **receive a PDF & your project back** showing a breakdown of your marks with comments
  - I will also address comments throughout your program

- If you disagree or believe I made a mistake, reach out within a week and we will resolve it
- I run plagiarism checking on my end every submission. I always catch at least two people on PA 1, please just ask for an extension if you need it

EXTRA CREDIT

- (In Lab) I will make questions that are difficult
  - It may be solving a problem
  - It may be telling me what the function will return given some input
  - 1-2 points, based on difficulty

- (In Lecture) Andy grades some assignments at the end of this semester
  - Be sure you submit them to Andy via email – I do not grade these
  - All extra credit must be sent directly to Andy
  - PA 9 (Submitted through Canvas)

RECOMMENDATIONS FOR SUCCESS

- Ask questions as they come up (whether you are in lab or lecture)
- Attend office hours when you do not grasp concepts
- Use whiteboards at home and in lab (will be very helpful)
- Work in groups
- Watch videos on specific topics
- Ask classmates or any other person you know who is good with these concepts

Do not:

- **Copy/paste from friends or the Internet**
- **Copy from the Internet without understanding the content**

RECOMMENDED WEBSITES

- Documentation
  - cplusplus.com
- Tutorials
  - w3schools.com
  - tutorialspoint.com
  - geeksforgeeks.org
  - Youtube video
- Use dorking to search if you find a site you like
  - i.e., `atoi site:cplusplus.com` <= Will search for atoi only on cplusplus.com

STACK OVERFLOW

- It is good and bad
- Sometimes it will give you a good answer with actual reasoning
- Other times you will get the correct answer, but with no or wrong reasoning

- Always make sure you understand what you use
  - **If you cannot explain it, don't use it**
  - Always cite the source. *Even if it is a single line*

QUESTIONS?

INTRODUCTIONS

- Name
- Degree & class standing. What made you choose it?
- Experience with code (What languages, if any)

LAB #1 – SECTION 3

GitHub – Watch my Video (please find it on **kyleparker.me**, click on **Projects**)

SIGNUP AT GITHUB.COM

- You can either create a student account and gain access to some features:
- https://education.github.com/pack

GH & CANVAS WILL HOST SLIDES AFTER LAB

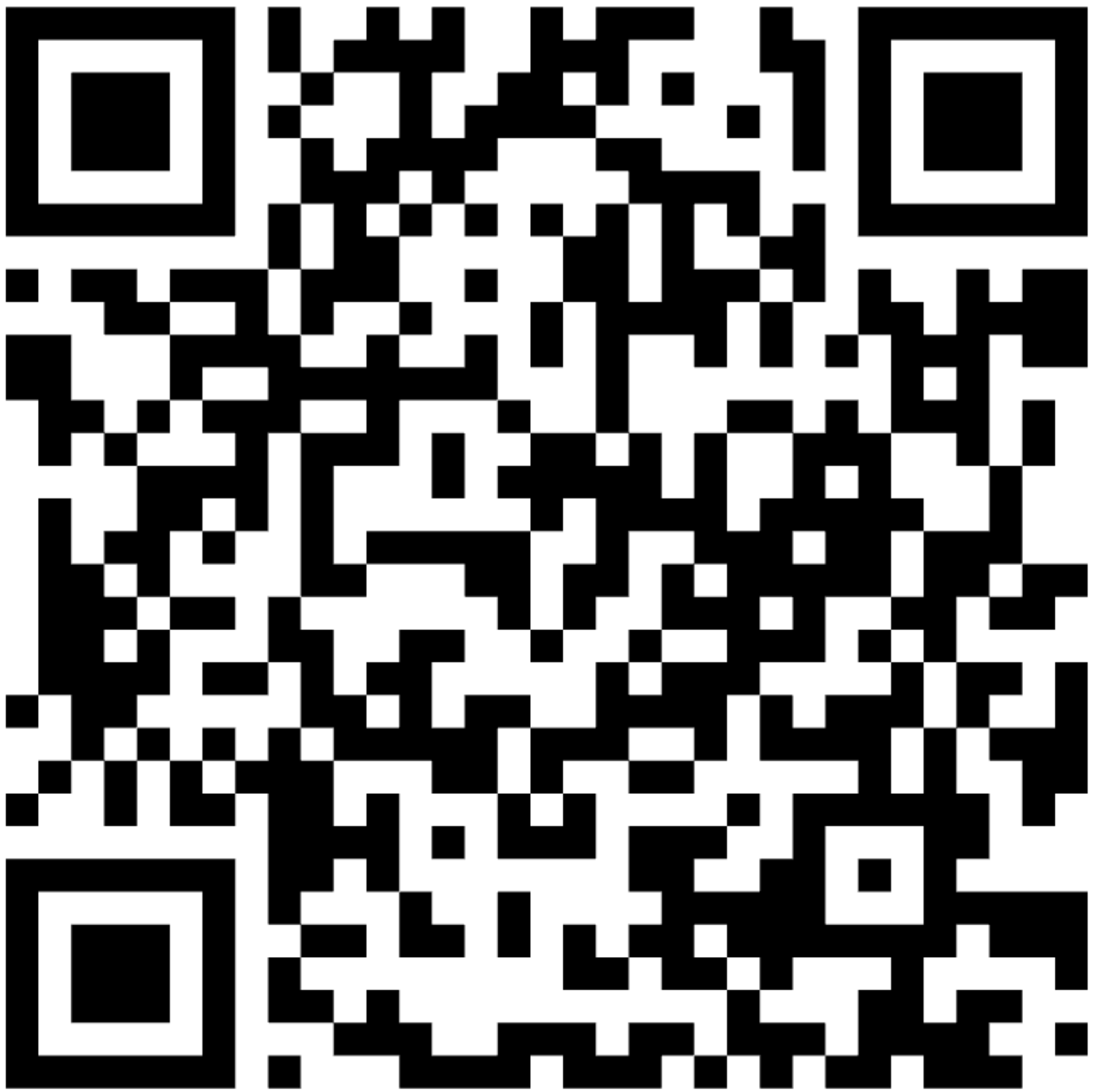
- I will post all content from labs to GH
- I will post review material to GH
- I will post any code we go over on GH
- I will post request content on GH
- Depending on time, I may provide some starter code for a lab (it will be advanced) and you will build from it

- This does not mean you must create an account, but I **strongly** encourage it
  - It will help with the final project in this class!

LAB REPO

- https://github.com/swiftlydesigner/122-Practice-Review

- Feel free to share with friends!



QUESTIONS?

LAB #1 – SECTION 8

C Review

DYNAMIC MEMORY

- What function do we use to allocate memory?
  - malloc (malloc, calloc, realloc)
  - `void * malloc(size_t size);`
  - `void * calloc(size_t count, size_t size);`
  - `void * realloc(void *ptr, size_t size);`
- What function do we use to deallocate memory?
  - free
  - `void free(void *ptr);`
- What is the difference between malloc and calloc (no, not the params)?
  - In calloc, "`The allocated memory is filled with bytes of value zero.`"

MAN COMMAND – *NIX

- Can you find out how to use mallow without internet on a *NIX (Linux, macOS, etc.) system?
  - Yes!
    - Linux or macOS users: try using it today! You will be required to use it down the road.
  - You use the command `man`

CHECK YOUR UNDERSTANDING

- What will line contain after the following call?
  - DATA.DAT (DATA_FILE): `Hello, World!`
  - VAR DECLARTION: `char line[10];`
  - CALL: `fgets(line, 10, DATA_FILE);`
  - Now, line is:
    - Hello, Wor

CHECK YOUR UNDERSTANDING

- Bubble Sort Steps:
    1. Get length of array (or object used) (let's say MAX)
    2. CUR = 0
    3. Check CUR with CUR + 1 (SET CUR = 0 first round)
        1. If CUR > CUR + 1, swap [Create a function which can handle any type!]
    4. ++CUR
    5. Repeat [3] and [4] until CUR + 1 == MAX (we can also say CUR == MAX – 1)
    6. --MAX
    7. if MAX > CUR + 1 then GOTO 3
    8. Sorted! (After $O(N^2)$)

CHECK YOUR UNDERSTANDING

- What is Guard Code?
  - #IFNDEF MY_HEADER_H
  - #DEFINE MY_HEADER_H
  - // Whatever you need in your header
  - #ENDIF /* MY_HEADER_H */
- Why do we like it?
  - It is structured
  - It prevents circular references
    - It prevents duplicates

RECURSION

- A function or method that could call itself at least once
- Could increase the performance on some devices
- Increases memory usage which could decrease performance
- Makes some code easier to write (compared to loops)

BIT MANIPULATION

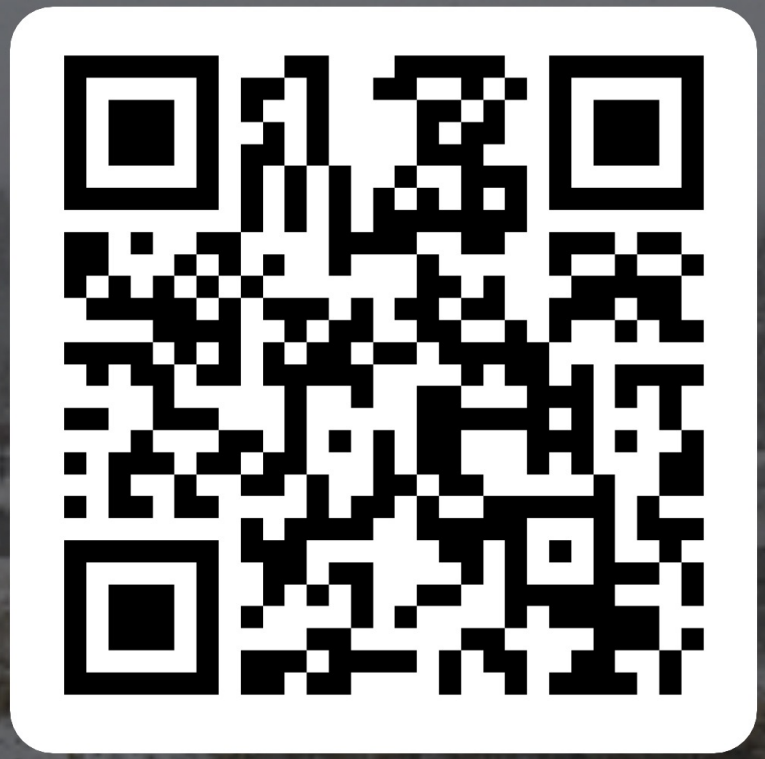
- EXCLUSIVE OR (XOR) ^
  - **One or the other is 1**
  - Exactly 1 bit is 1
- OR (OR) |
  - **At least one bit is 1**
- AND (AND) &
  - **Both bits are 1**
  - Exactly 2 bits are 1

I1	I2	AND (&)	XOR (^)	OR (|)
0	0	0	0	0
1	0	0	1	1
0	1	0	1	1
1	1	1	0	1

ATTENDANCE

- Enter Code:
  - IDA

CS 122 L3 - Lab 1 - Spring '25



PROBLEM 1 (30 MIN)

a. Write a function that recursively reverses a string. Recall, a recursive function is a function that calls itself. These functions have at least one base or simple case and at least one recursive step. The base case(s) have known solutions. As the function is called, the problem is broken down into simpler parts that are closer to the base case.

1. Set a to the first char in str
2. Pass from after a in str to itself
3. Insert a at the end

PROBLEM 2 (50 MIN)

Find test cases on my GH.

b. Write a function called `myStrTok()` that behaves in the following manner (taken from http://en.cppreference.com/w/c/string/byte/strtok). Note: all references to `strtok()` should be replaced by `myStrTok()`. You will need to declare a *static* pointer inside of `myStrTok()` to track tokens through successive calls!!! For example:

```
static char *pToken; // static will allow for the
                     // pointer contents/direct value to retain
                     // changes through successive calls
```

PROBLEM 2 CHALLENGE

- If you finish all problems, write a function for strtok_s
- strtok_s is the thread-safe version of strtok
  - this means there is no static variable within strtok_s

- Threads
  - A way of running code "concurrently"
    - Practically, only X programs run at the same time, based on X processors (4 cores = 4 programs running at once
    - If you have 100 programs and 10 cores, then they will switch between contexts and only 10 run in parallel. More on this in CptS 360!

PROBLEM 3 (60+ MIN)

- The first solution should be very easy
- The second one should be moderately easy
- The third has a few ways to approach it

c. Write a function that integrates or merges two unsorted strings into sorted order. You will need to provide multiple solutions.

i. Solution 1: Merge items into a third, fixed-sized array.

ii. Solution 2: Merge items into a third, dynamically allocated array, which grows as each item is inserted. Consider using `realloc()`. You will find more information about `realloc()` at: http://en.cppreference.com/w/c/memory/realloc. We are NOT using a linked list to solve this problem!!!

iii. Solution 3: Merge items without the use of a third array.

QUESTIONS?

HANDLING ADJACENT COMMAS IN CSV FILE

1. Figure out if there are adjacent commas
2. Insert an invalid flag (-1, -, NA, etc.)
3. During parsing, check for the flag if it is not an integer
    1. Many ways to express this; I am using a structure in my example

Note: There are many approaches and in-between steps.