CPTS 122 L3

Boring Part – Course Guidelines

STRUCTURE OF LABS

- Work in groups of 2-4. If you do not work in your group, you will receive a 0 for the day
- Everybody must contribute (nobody is doing all the work or sitting back)
  - If you have coding experience, help your group and make sure everybody understands
- Use whiteboards to enhance thought processes and when you are stuck
- Demonstrate your lab to me upon completion (preferably part by part)
  - Work on the next part as you wait for me
- Take 2-3 5-minute breaks throughout labs – I do not regulate this, do it when you want
- Do not make a new project for every problem. Make a new C/H file for each problem or separate by comments (at minimum 3 files).

EXPECTATIONS

- Show up on time
- Be an active participant in your group (we will mix them up if people do not work)
- Help others who are struggling or stuck on a problem
- No watching videos on your phone
- Be reasonably loud (don't yell or be overly reactive to things)
- **Do not play games once you finish the lab**
  - Instead, work on: Helping others, PAs, Review Past Materials, Past Labs
- Do not leave until the end of lab
- **If you are sick, do not show up.** If you are sick or have a personal emergency, email me about your absence
  - Next lab you are here, show me your work for credit

GRADING (LABS)

- Based on effort
  - If you are on your phone, not working in a group, being obnoxious, or otherwise not working on the assignment, expect a reduced grade (0 in cases where no work is done, or others are distracted)
  - If you are working, actively helping those who are stuck, expect full points
- Complete 2/3 of the assignments
- Do not show up with completed exercises!!! You will get a 0 and a possible referral to Andy
- Extra credit will be available for labs as I have time to produce problems.
  - Worth 1-2 points depending on complexity
- Points put in Canvas within the hour. Note I may not put grades in until later that night due to schedule conflicts.

GITHUB

- I will post all slides I make to a GitHub Repo I designed for this lab.
- For those who do not know what GitHub is:
  - A version-control software (VCS) used to manage code remotely
  - Allows multiple people to work on a project
  - Allows developers to use branches to work on new features without overriding one another
  - More details in later classes (321, 322, many others)
- I recommend using it in this class as a backup in the case of a corrupt drive (has happened to two past students)
- **DO NOT USE IT TO PLAGIARIZE!!**

GRADING (PAS & QUIZZES)

- Typically graded within one week. I may get behind at specific times
  - In the case I get backed up, email me if you need feedback that week and an updated grade. *Your assignments may be graded by aux TAs during peak times*
- I will provide detailed feedback on all assignments. It takes me longer than other TAs since I go though and give in-line and summary feedback (for PAs).
- Quizzes are simple to grade, and you should expect feedback the following Saturday
  - I.e., you submit it Monday the 18th, grades should be returned on Saturday the 23rd
- If you disagree with my grading, **contact me first**, then go to Andy
- I am grading on a Mac – this means some things will not work the same (most of the time I change your code so it works correctly)

GRADING (PAS)

- If your assignment does not compile, then you will receive 60%.
  - Note: I may need to adjust your code being on Mac – there are few times I am unaware of how to bridge your code.
- To ensure it will run on my machine, go to:
  1. Build > Clean Solution > Run
  2. If it compiles on your end, email me and we will go over in office hours or lab
- Most common problems
  - Reading/writing outside of bounds in arrays (my computer may crash or not read the correct values)

CONTESTING GRADES (ALL ASSIGNMENTS)

- Talk to me in lab, send me an email, or come to office hours
  - If you email me, we will still talk in person most of the time
- I am very easy on adjusting grades
- ***I make mistakes***, tell me which problem you believe there is a mistake
- When we have exams (excluding final), bring any mistakes up in person
  - Only adjustments will be if I mismark your answer
  - I grade these late at night typically and make 2-3 mistakes for the whole lab throughout the semester
  - Must be done in person – this will not be negotiable over email

EXTENSIONS (PAS ONLY)

- Given on all PAs
  1. Send me an email, provide a reason (personal reasons do not need to be in detail)
  2. You can assume I will give it unless you have requested it on every assignment and fail to submit it on time
- Email me before the deadline, if it is due the 10th, do not email me on the 11th, you will get -10% (with few exceptions)
  - You can even request one on the 8th if you are unsure about completing it on time. No shame in getting an extension and not using it.
  - Once I email you and approve the extension, put a comment in your assignment if I have not gotten to it.
- **No extensions are given on quizzes.** *Note: I do not download submissions until the morning*

PLAGIARISM (1 OF 2)

- Using any content from the Internet, a friend, or something/one who is not you
  - If you use sources from the Internet, cite them; this should only be for features such as a more attractive CLI. **If you cannot explain it, then it is considered plagiarism**
- If you change a word or two from a source with or without a citation
- If you send a friend your PA(s), that is plagiarism
- If you do plagiarize and get caught, be honest – lying will make it worse and you are more likely to fail the course with a permeant mark on your transcript
  - No body wants to hire someone who can copy/paste, AIs are free and can easily to that
- If you are caught, you will receive a 0 for that assignment or quiz problem
  - All instances are reported to Andy – if I refer you to him, he will expect you

https://communitystandards.wsu.edu/policies-and-reporting/academic-integrity-policy/ for WSU's Policies regarding Plagiarism

COMMON REASON FOR PLAGIARISM: RUNNING OUT OF TIME (2 OF 2)

- Yes, many people do this
- **If you are low on time, ask for an extension** – not your friend for the solution
- No, I am not lenient on plagiarism
- Yes, you will get a zero
- **If you are stuck on a problem, ask me, another TA, Andy, or have a friend *explain* how to do it. Do not let them send you the PA. If they do, delete it and ask them to explain the work in person or find another way.** If you need more help, ask a TA, myself, or Andy

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), 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 a 50-100% point reduction
- Be sure that everybody involved can write the code for the objective

IDES

- Windows
  - Visual Studio
- macOS
  - Xcode (I prefer this over Boot Camp or running a VM for Windows)
    - More rigorous debugging tool with ease of use
- Windows, macOS, Linux
  - CLion (Free if you signup with your WSU account) [I have not personally used this]
- Linux
  - gcc and gdb (cli tools – advanced)

OFFICE HOURS

- Dana 134
- Day: TBA
- Hours: 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 break.

CONTACT

- 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)
- Include “CPTS 122 L3:” in the subject
- Contact me for:
  - Setup appointments
  - Questions/clarifications on assignments
  - Missing a lab
  - Missing an exam (Attempt to contact Andy as well or first)
    - Limited acceptable reasons.

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 videos
- 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*

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 – I do not grade these

QUESTIONS?

INTRODUCTIONS

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

LAB #1 – SECTION 3

GitHub

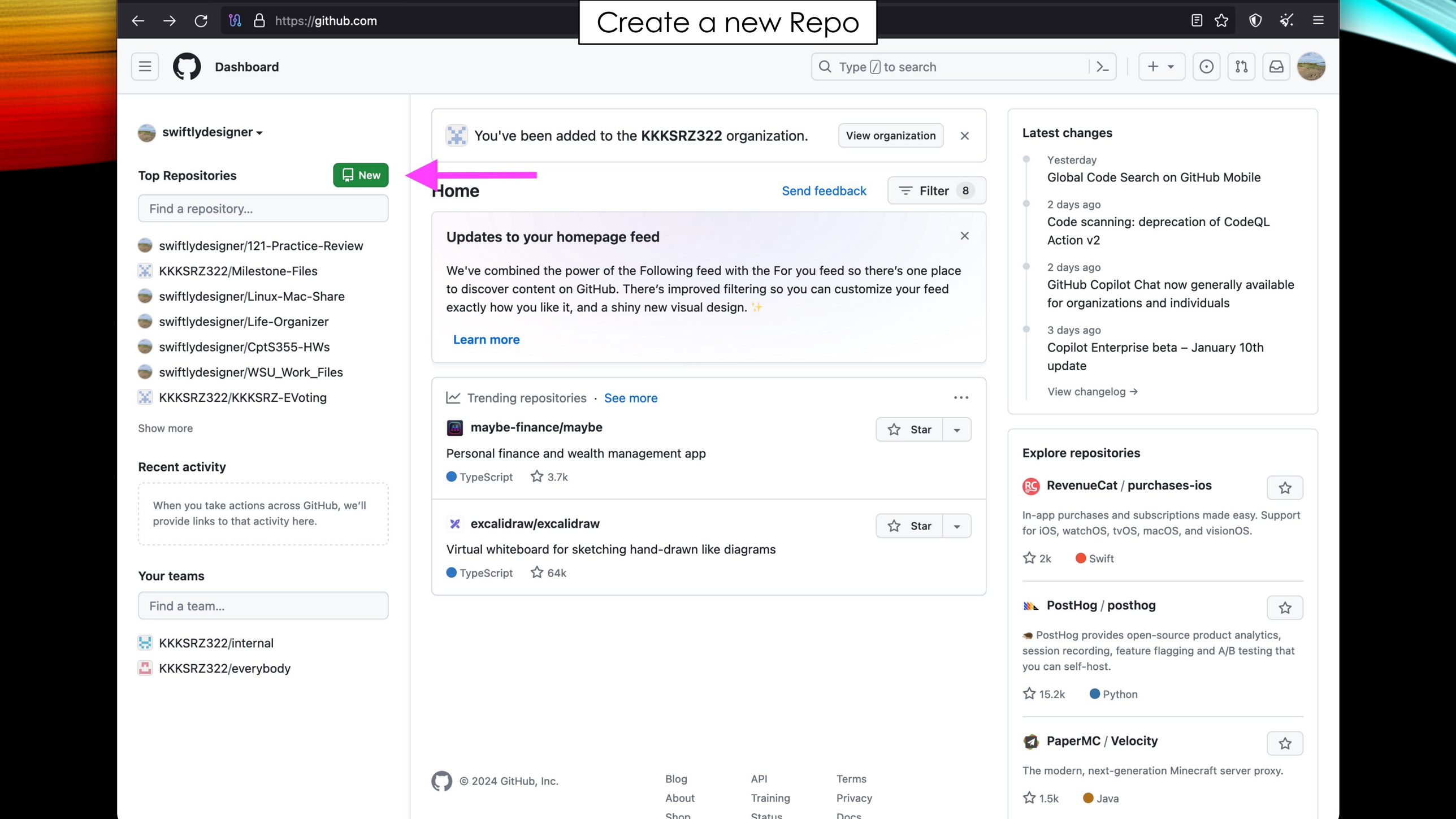
SIGNUP AT GITHUB.COM

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

WE WILL USE GH FOR ALL SLIDES

- 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!


Create a new Repo
https://github.com
Dashboard
Type / to search
swiftlydesigner
Top Repositories
New
Find a repository...
swiftlydesigner/121-Practice-Review
KKKSRZ322/Milestone-Files
swiftlydesigner/Linux-Mac-Share
swiftlydesigner/Life-Organizer
swiftlydesigner/CptS355-HWs
swiftlydesigner/WSU_Work_Files
KKKSRZ322/KKKSRZ-EVoting
Show more
Recent activity
When you take actions across GitHub, we'll provide links to that activity here.
Your teams
Find a team...
KKKSRZ322/internal
KKKSRZ322/everybody
You've been added to the KKKSRZ322 organization.
View organization
Home
Send feedback
Filter 8
Updates to your homepage feed
We've combined the power of the Following feed with the For you feed so there's one place to discover content on GitHub. There's improved filtering so you can customize your feed exactly how you like it, and a shiny new visual design. ✨
Learn more
Trending repositories · See more
maybe-finance/maybe
Star
Personal finance and wealth management app
TypeScript 3.7k
excalidraw/excalidraw
Star
Virtual whiteboard for sketching hand-drawn like diagrams
TypeScript 64k
Latest changes
Yesterday
Global Code Search on GitHub Mobile
2 days ago
Code scanning: deprecation of CodeQL Action v2
2 days ago
GitHub Copilot Chat now generally available for organizations and individuals
3 days ago
Copilot Enterprise beta – January 10th update
View changelog →
Explore repositories
RevenueCat / purchases-ios
In-app purchases and subscriptions made easy. Support for iOS, watchOS, tvOS, macOS, and visionOS.
2k Swift
PostHog / posthog
PostHog provides open-source product analytics, session recording, feature flagging and A/B testing that you can self-host.
15.2k Python
PaperMC / Velocity
The modern, next-generation Minecraft server proxy.
1.5k Java
© 2024 GitHub, Inc.
Blog
About
API
Training
Terms
Privacy

Provide Repo Details

https://github.com/new

New repository

Type / to search

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * swiftlydesigner / **Repository name *** 122-Practice-Review

122-Practice-Review is available.

Repo Name. Must be unique within your Organization or Account

Great repository names are short and memorable. Need inspiration? How about **improved-carnival** ?

Description (optional)

Assistive Material for CptS-122 at WSU

Choose when you want to share with the world. i.e., create OSS (Open-Source-Software)

- **Public** — Anyone on the internet can see this repository. You choose who can commit.
- **Private** — You choose who can see and commit to this repository.

Choose when you want to work in a group but you do not want the world to see

Initialize this repository with:

- [ ] **Add a README file**

This is where you can write a long description for your project. Learn more about READMEs.

Add a README so you can describe your repo

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

This is a set of rules to ignore certian files. i.e., ignore generated files (a.out). Can manually grow and shrink this list.

I chose MIT License since it fit best. Always look at the different Licenses as the best for you may vary

Choose a license

License: MIT License

A license tells others what they can and can't do with your code. Learn more about licenses.

You are creating a public repository in your personal account.

Default New Repo

swiftlydesigner / 122-Practice-Review

Type / to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

122-Practice-Review

Public

Pin Unwatch 1 Fork 0 Star 0

main 1 Branch 0 Tags Go to file t Add file Code

swiftlydesigner Initial commit fec65c3 · 8 minutes ago 1 Commits

LICENSE	Initial commit	8 minutes ago
README.md	Initial commit	8 minutes ago

README MIT license

122-Practice-Review

Assistive Material for CptS-122 at WSU

This is the default when you Select:

Git Ignore: None

License: MIT License

INCLUDE README

About

Assistive Material for CptS-122 at WSU

Readme
MIT license
Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

LAB REPO

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

- Feel free to share with friends!

QUESTIONS?

LAB #1 – SECTION 3

C Review

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)

PROBLEM 1

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

Read the directions. I also provided sample test cases on GH. Please advise those if you need help!

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

These are straight-forward. It can be tricky to do these up front as the code may not seem as straight forward as the algorithm.

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?