USF UNIVERSITY OF SAN FRANCISCO
CHANGE THE WORLD FROM HERE

Getting Help

CS 272 Software Development

Professor Sophie Engle
Department of Computer Science

Managing your time...

- **Goldilocks Principle**: Must be "just right"
  - Need enough time to **get into the flow**
  - Need enough flexibility **to take a break**
- Want to maximize **productivity**
  - Avoid having to program when cannot be productive
- Plan **2-3 hour** chunks **2-4 times** a week

Getting started...

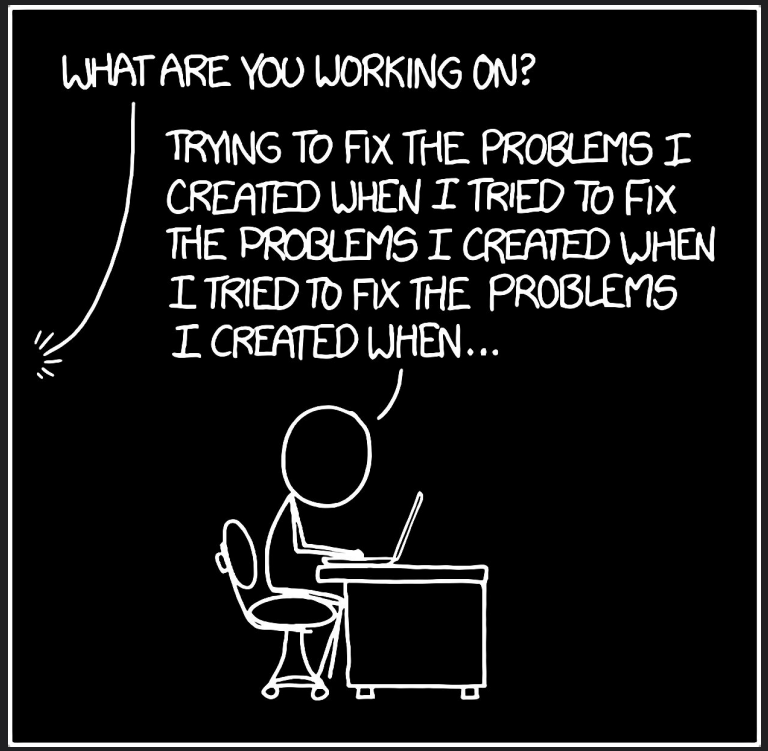
- **RT_M*** as a starting point
  - *Learning to read documentation is another skill!*
- Focus on a **single method** or subset of functionality
- Focus on a **single test** or group of method/subset
- Output the **parameters** passed in by the test(s)
- Look at what was **expected** for those parameters
- Build code up **iteratively** and test as you go

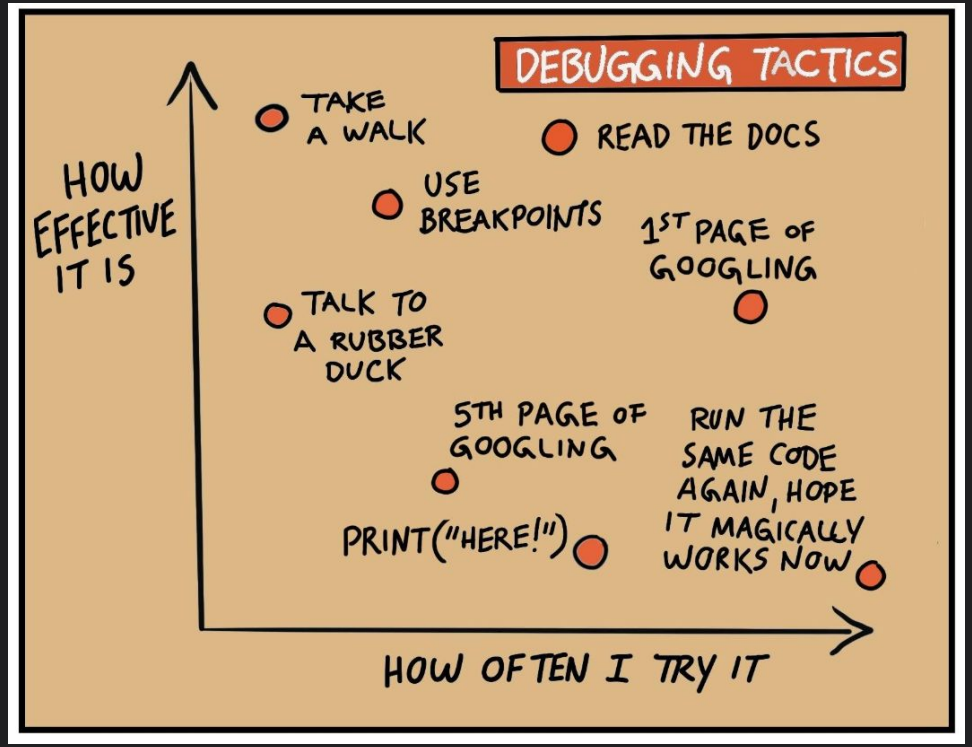
**RT_M: Read the ... manual*

Making progress...

- Programming progress is **unpredictable**
  - Even first homework varies due to many factors (often not skill-based)
- Programming progress can *easily* come to a **halt**
  - **Confused** and unable to make progress
  - **Stuck** on a bug and unable to make progress
  - **Exhausted** and unable to make progress*

**Brains get tired too! When your brain needs rest, no amount of "work ethic" is going to help.*





Left: xkcd by Randall Munroe, https://xkcd.com/1739/
Right: FaaS and Furious by Forrest Brazeal, https://faasandfurious.com/71

Debugging Strategies

- Try to understand the problem before trying to solve it.
    - *Syntax? Compile? Runtime? Exception? Output?*
- [illegible]gure out what you[illegible] co[illegible]e is d[illegible] (debugger, print[illegible]).
- B[illegible]a[illegible] o[illegible] c[illegible]e[illegible]nter[illegible]ally [illegible] t[illegible] be[illegible]vi[illegible]
- S[illegible]le[illegible] o[illegible] e[illegible]oble[illegible] and a[illegible]oun[illegible] f[illegible] c[illegible] fo[illegible]is[illegible] d
- Explain your code line-by-line to a rubber duck.
- Take a *productive* break to give your mind a break.

Don't Panic

Getting stuck...

- Work on a problem for 30 minutes to 1 hour
  - Explore how your code is *actually* running
  - Narrow down where to focus your attention
  - Focus on the next step not the final solution
- **If stuck on a problem for over an hour, ask for help!**
  - *Worst Case:* You get unstuck before we can help (yay!)
  - *Best Case:* We help you get unstuck (also yay!)

Getting help is...

- Getting programming help is a **skillset**
  - It isn't something you know (i.e. memorization), it is something you do (i.e. **requires practice**)
- Important both for school and future career
  - Problem solving and debugging often requires a **different perspective** than the one you have

Getting help is...

- Not as simple as just **asking for help**
  - Helps to ask the right type of questions
  - Helps to have put in initial effort (**30 min to hour**)
- Not as simple as just **receiving help**
  - Helps to get the right type of answers
  - Providing help is another skillset (often untrained)

When getting help...

- Goal is to **get unstuck** and solve the problem yourself
  - *Avoid:* No growth to your understanding or skillset
  - *Prefer:* Growing your understanding and skillset
- Goal is still to **do the work** of problem solving
  - *Avoid:* Not being able to contribute to "the work"
  - *Prefer:* Helping each other do "the work"

Getting unstuck...

- Post a **LINK** to your code on GitHub – *no screenshots*
- Tell us the **FIRST TEST** your code is NOT passing *(this should be your focus)*
- Confirm tests before that ARE passing
- Give us the **TEST OUTPUT*** (can copy/paste from Eclipse)
- Tell us what you have tried, so we don't suggest something you have already done

**Output includes console output and any stack traces!*

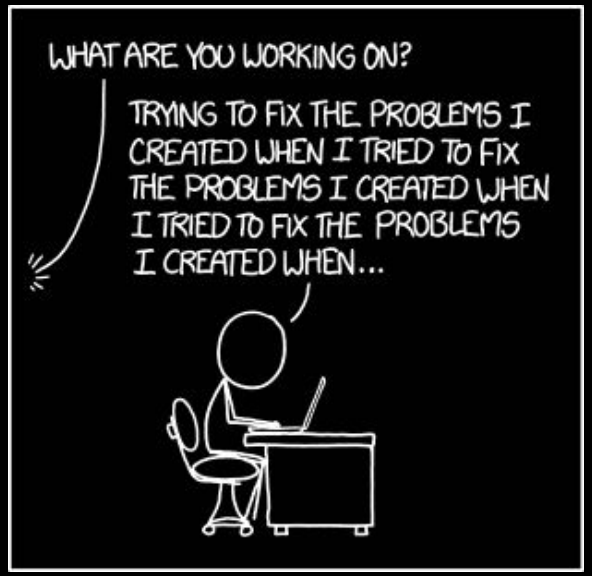
Keep in mind...

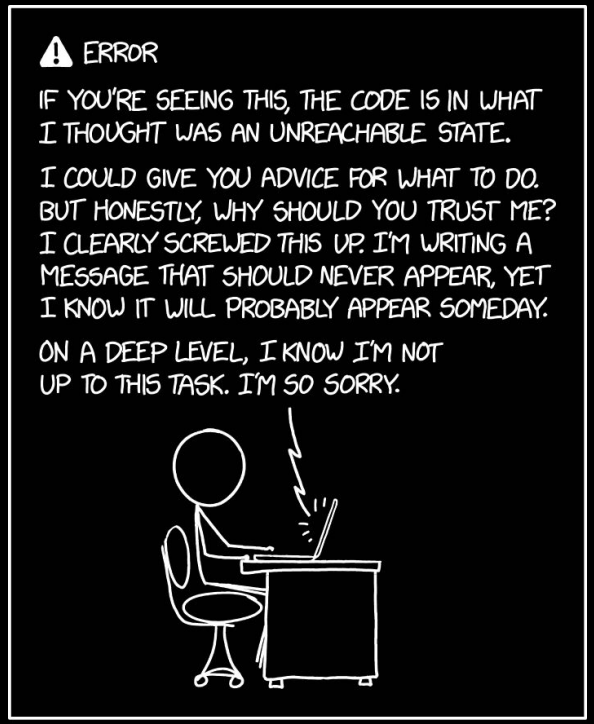
- The problems we tackle should be **solved iteratively**, regardless of whether you are stuck
- You **need practice** with both problem solving and getting help with that process
  - *OK:* You don't know how to ask the right questions
  - *NOT OK:* Not asking for help if you are stuck!

Keep in mind...

- You need **you** to write a search engine
  - You don't need **the solution** to this problem
  - You need **the practice** solving complex problems
- Don't be afraid to **use the force** (subconscious)
  - Try things until something clicks
  - Take a break when not making progress

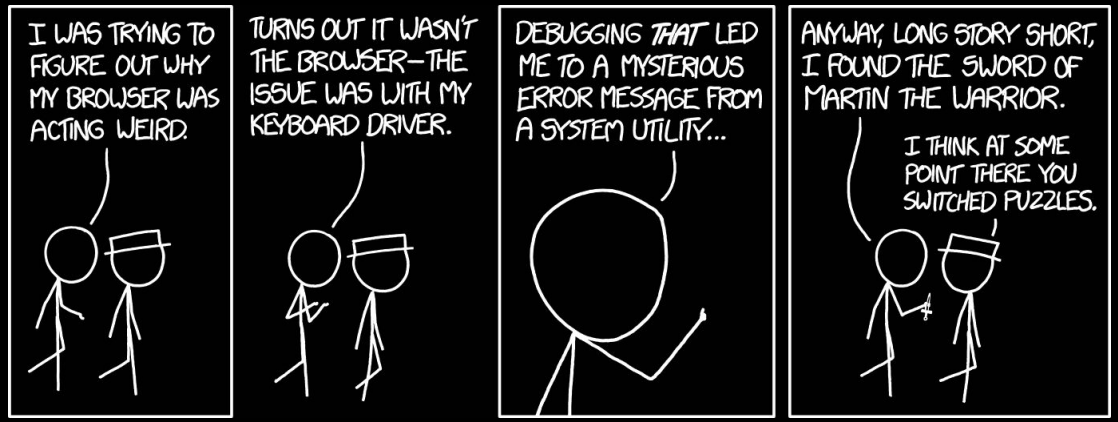
Questions?





NEVER WRITE ERROR MESSAGES TIRED.

Left: https://xkcd.com/1739/ · Middle: https://xkcd.com/1906/ · Right: https://xkcd.com/2200/



Left: https://xkcd.com/1722/ · Right: https://xkcd.com/979/