

Getting Help

CS 272 Software Development

Getting Programming Help

- Getting 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 sometimes requires a **different perspective** than the one you have (unavoidable)



Getting Programming Help

- Not as simple as just **asking for help**
 - Helps to ask the right type of questions
 - Helps to have put in 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)



Why Ask Questions?

- Goal is not to ask for (or provide) "the solution"
 - *Avoid*: What is the answer for [x]?
 - *Better*: **How** do I find the answer for [x]?
 - *Better*: **How** do I get started solving [x]?
 - *Prefer*: What should I **try next** to solve [x]?



Why Ask Questions?

- 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"



How to Get Started (with Tests)

- **RT(F)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(F)M: Read the (...) manual*



How to Get Help (with Tests)

- Work on a problem for 30 minutes to 1 hour
- **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 and you can keep making progress (also yay!)
- Focus on the next step not the final solution



How to Get Help (with Tests)

- Post a LINK to your code on Github – *no screenshots*
- Tell us which tests your code is passing
- Tell us the FIRST TEST your code is not passing
(*this should be your focus*)
- 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



Time Management

- 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.*



Time Management

- **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



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** (i.e. take a break?)



Questions?

