CS 351: Data Structures and Algorithms

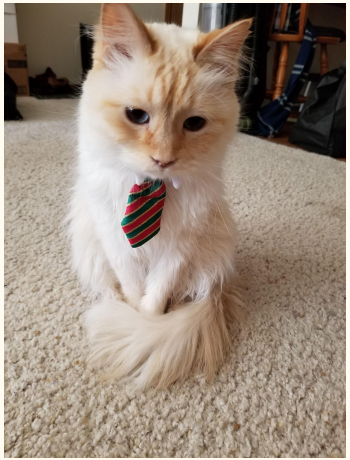
—

Week 1

Introductions

Natasha

- BFA in Film/Animation
- Currently pursuing MS in Computer Science, focus in AI
- President of Women in Computing
- Interned at NASA Kennedy Space Center
- Will do AI/ML research at MIT this summer
- 2 cats
- Hobbies/Interests: watching movies, reading, painting, hiking, rock climbing, science fiction/fantasy, experiencing new things



Important Notes

Homework

- Homework is available at 10pm on Mondays
- Github/Github Classroom
  - Make sure you commit often and push at least at the end of each time you work
  - Push to submit in progress/finished work
- Homework is due at 10pm on Mondays, NOT MIDNIGHT
- Graded on accuracy (pass tests), style, and showing work
- 1 homework grade is dropped

Make sure your homework doesn't have any compiler errors (red x by the file name) and doesn't have any compiler warnings (yellow triangle by file name)!

Make sure you "commit" your work frequently and "push" after each coding session!

DO NOT share code or look at each other's code--this is NOT actually helpful for you to learn, and will be detrimental on exams!

Make sure you put a comment at the top of your homework if you discuss the homework with other students or seek help from the tutoring center. Include names! This is like citing references!!

How to Approach Homework

- READ the handout
- Read the general section (intro) of the related chapter in the book
- Read any relevant parts in the rest of the chapter
- Run UnlockTests.java
  - This will help you UNDERSTAND the assignment and will make coding A LOT EASIER
- ALWAYS start with the invariants for HW2 and later
  - Make sure there is nothing wrong with your data structure
  - EXTREMELY HELPFUL for debugging
- FINALLY start writing code

Lab

- Labs available 5pm Wednesdays
- Attendance is required
  - Speak to me prior to lab time if you have a conflict
- Partial credit for attending first half
- Full credit for completing lab
- If you do not finish during lab:
  - Finish before 10pm Friday
  - Make sure to PUSH your changes
  - Email me to let me know you have finished
  - If there was a written part of the handout, attach a picture to your email!
- 2 labs will be dropped

Exams

- Midterm is the week before Spring Break
  - Tuesday lecture: Reading/Understanding code
  - Thursday lecture: Writing code
  - First half of lab: Concepts short answer
  - Second half of lab: Debugging
- Final exam: May 14th, 7:30 am
  - Contains same sections as midterm, but a bit shorter
- Practice exams posted on course site
- Study homework solutions!!!
- Questions in the book are very good for studying!

Academic Resources

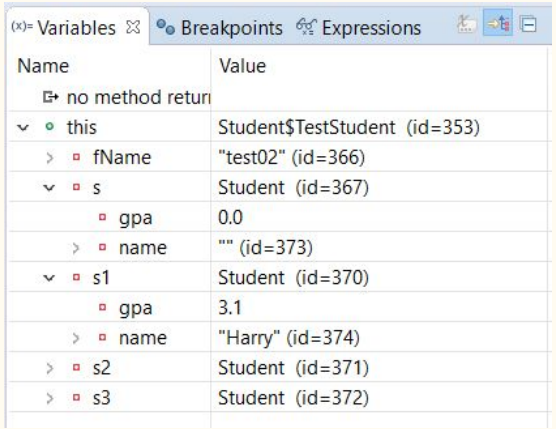
- Post homework questions on Piazza!
  - https://piazza.com/uwm/spring2020/compsci351/home
  - See if question has already been asked
  - Can post anonymously
  - DO NOT SHARE CODE
  - Opportunity for extra credit if you provide help to others: do NOT share code or tell them what to do--give hints and ask questions to guide their thinking! You don't learn by getting answers!
- CEAS Tutoring Center, EMS E280
  - 8am to 7pm, Monday - Thursday
  - https://uwm.edu/engineering/current-students/tutoring/
- Office Hours
  - Posted on course site and syllabus
  - Monday 3-5pm, EMS E218 or E280
- Q&A page on course site: issues with Eclipse/Git, general issues
- Oracle Java Documentation
  - https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html

Non-Academic Resources

- Technical Interview Practice
  - www.hackerrank.com
  - www.leetcode.com
- Google Interview Process (similar for most SWE internships)
  - https://docs.google.com/presentation/d/1_6c6eu1oaDcJeKGcu43wtal8OeFNL6xMmmoSiDt9l5A/edit
- Mental health resources/stress counseling:
  - https://uwm.edu/mentalhealth/
- Accessibility Resource Center:
  - https://uwm.edu/arc/
- UWM Food Pantry:
  - https://uwm.edu/studentassociation/uwmfcp/
- Many student groups
  - Women in Computing
  - IEEE-Computer Society
  - Google Developers Group

Lab 1: Debugging

Debugging Tools



- Breakpoints
  - Set point for execution to stop
- Step In
  - Goes into a method call
  - Goes to next line
- Step Over
  - Goes over a line
  - Does not go into a method call
- Step Return
  - Returns to where the method was called
  - Method completes evaluation, but you skip ahead instead of through each line
- Variables Window
  - Allows you to see all of the variables in that instance
  - See what variables are being updated and when

How to Approach Debugging

- Identify tests that are failing
- UNDERSTAND why these tests are failing and what they are testing
  - Read the failure trace and the test
  - Draw pictures of data structure and how it should behave
  - Put a breakpoint on the line causing the failure
  - Go through step by step and see how the state is changing with the Variables pane
  - Draw pictures of what your code is doing
- Make sure you are not writing code just to pass tests--write code to achieve the correct behavior!
  - If you end up with a lot of if statements, you are doing something wrong!

In class exercise