LEC 17

CSE 122

Third Party Libraries

Questions during Class?

Raise hand or send here

sli.do #cse122

BEFORE WE START

Slido vote & talk to your neighbors:

What has been your favorite assignment so far this quarter? *Could be for CSE 122 or other class!*

Music: 122 26Wi Lecture Tunes

Instructor: Adrian Salguero

TAs:

Ava	Dalton	Neal	Shreyank
Blake R	Dani	Neha	Sthiti
Blake P	David	Nicolae	Sushma
Cady	Diya	Nicole	Suyash
Caleb	Hanna	Rio	TJ
Cole	Ivy	Rohan	Wesley
Colin	Mahima	Saachi	Yang
Connor	Medha	Shreya	

Lecture Outline

- Announcements
  - Final Exam Logistics
- Third Party Libraries
- Open-Source Software
- Intro Creative Project 3

Announcements

- Creative Project 3 will be releasing later today!
  - Final assignment for CSE 122!
- R6 due on Tuesday, March 10th at 11:59pm
  - Eligible assignments: P2 and C2
- Final Exam details will be posted tonight on the course website!

Final Exam Logistics

Important Points:

- Tuesday March 17th from 12:30pm – 2:20pm in KNE 130/220 🍀
- There will be assigned seats (posted soon)
  - If you would like a left-handed seat, you will be able to request one through the form on Ed.
- Don't cheat!
  - Do not start early or work after time is called
  - No electronic devices or communicating with others during the test
- You will be able to bring in one reference sheet of your own notes (just like quizzes)
  - Additionally, we will provide a reference sheet during exam (posted on website soon)
- Topic list
- See website for details about policies w.r.t. what we expect of your answers

Lecture Outline

- Announcements
  - Final Exam Logistics
- **Third Party Libraries**
- Open-Source Software
- Intro Creative Project 3

Library

- The term **library** or **third party library** is used broadly to refer to a collection of code someone has written, packaged up, and shared for others to use
  - Example: JUnit is a library!
- There are a lot of libraries out there!
  - Here is just one person's list of favorites with only 687 different libraries 😅
- Example: Our very own course website!

Lecture Outline

- Announcements
  - Final Exam Logistics
- Third Party Libraries
- **Open-Source Software**
- Intro Creative Project 3

Open-Source Software

- Libraries are just code that people write! You can make your own if you want! Or contribute to one that wants help!
- **Open-Source Code** are projects where all the code is visible to the world.
- Many Open-Source projects are also run by teams of individuals all working on the same code base. Need collaboration tools to make it easy for many people to work on one piece of code.
  - Wanna try? Check out https://www.firsttimersonly.com/
- Collaboration Tools
  - git to manage the process of collaborating on code
  - Websites such as GitHub or GitLab to host and share the code
    - Do **not** post solutions to your assignments publicly on these (or other) sites
- Won't focus on collaboration tools in the intro sequence
  - CSE Majors: CSE 391 Other Majors: CSE 374

Linus Torvalds
inventor of Git and Linux

Public != Free to Use

- Just because some code is publicly available to look at, doesn't mean you have permission to download, run, copy, or edit that code.
- All code should have a **License** or **Software License**, a legal set of permissions given by the code authors to users of the code
  - You need to make sure you are abiding by any licenses in code repositories you are interested in using (they are legally binding)
  - If you make public repositories and you want people to use it, you need to include some license.
    - The default license means nobody can use/copy/modify your code (and this gets very messy when you have other people working on the project)
    - How to choose a license (a common default is the MIT License)

Lecture Outline

- Announcements
  - Final Exam Logistics
- Third Party Libraries
- Open-Source Software
- **Intro Creative Project 3**

Design Patterns

- There are often common patterns / principles for the good design of code to make it
  - Easier to understand
  - Easier to write new features
  - Easier to test and spot bugs
- One of the most universal patterns is **abstraction** or **separation of concerns**
  - **Non-programming Example**: Running a restaurant would be much harder if every employee had to constantly switch being wait staff, chefs, bartenders, cleaners. Separating roles makes the whole job easier and efficient.
  - **Programming Example**: We've talked about the importance helper methods add to readability

- Optional Further Reading: Model-View-Control Design Pattern

Separation of Concerns

- Most applications separate the following things:
  - How the user views and interacts with an application
  - The “core logic” behind how the application works
- Example: Snapchat released a browser app a few years ago
  - Very unlikely they wrote the entire code base from scratch.
  - Instead, probably already had separated:
    1. The “core logic” of sending/receiving pictures, marking them as viewed, etc.
    2. A view on that controller that displays buttons, pictures, on an iPhone for the user to interact with
  - Now, adding a browser app doesn’t require any changes to the first system, just making a new version of the view to work on browsers instead of iPhones (and they probably have a different version for Android)
- Common theme: Separate the “logic” of an app from how the user interacts with it

Separation of Concerns in CSE 122

- With the introduction of Objects and Object-Oriented Programming, we can do this much more easily
  - If you wanted to make a visual app for Program Linting, you don't have to change most of the files! Just make a new LinterMain
- However, most of our previous assignments do this quite poorly
  - Example: For TODO List, you had code that interacted with the user mixed up with the logic for managing the TODO list! Concerns not separated!
- So Creative Project 3 is you deciding how to fix that!

Creative Project 3 – OOP It!

- Take one of our earlier assignments and rewrite it to use Object Oriented Programming to better separate the concerns of interaction and managing the state of the application
- You pick *one* of the assignments
  - P0: Stonks
  - C1: TODO List
  - P1: Music Playlist
- You get to choose which assignment you want to make OOP'ed, and how to design that object!
  - Externally it should have the same behavior as original assignment